M 2920 Pete Eaton (WB9FLW,2970) 10/12/85 3:08 PM L:125
KEYS:/COMPUTER TRADER MAGAZINE (CTM) ANNOUNCES PACKET SWEEPSTAKES/
TO: (Group 95), WB9FLW

To: All
From: Pete WB9FLW
Subject: COMPUTER TRADER MAGAZINE (CTM) PACKET $weepstakes Rules
Dist: Open
Note: Below is the exact text of a flyer distributed by CTM

In an effort to promote Packet Radio, CTM is sponsoring a ham contest like nothing you've ever seen. This contest is open to all licensed amateurs worldwide and boasts a prize list such has never been offered before anywhere!

In actuality, there are two separate contests involved in the CTM Packet $weepstakes. The first is what we affectionately refer to as the CTM Packet 50, which consists of connecting to all 50 U.S. States. The second is called the CTM Packet 100, which involves connecting to 100 or more countries.

The rules are simple. We don't want to play keep away with these prizes, we feel that it's going to be hard enough as it is.

1. All contacts must be made using packet radio on any legal amateur frequency

2. No store and forward contacts will be allowed. This means that you cannot use BBS or the store and forward capabilities of PACSAT to make a valid connection. All contacts must be live two way communications. Any number of digipeaters or gateways will be permitted.

3. All contacts must be made from the same state and country. We wouldn't want somebody traveling to all these places just to get the goodies now, would we?

4. All contacts must be confirmed via standard QSL type cards. All QSL cards must be signed by their operator, and contain the following information:

   Date and time of QSO (preferably in GMT).
   Operator's name, address, city, state, zip code and county.
   Must state that the contact was made using packet radio.
   Manufacturer of Terminal Node Controller used to make the connection (his TNC, not yours).

5. All contacts must take place after 0000 GMT on December 1, 1985.

6. Winner will be the earliest postmarked entry received that meets the contest criteria. Keep in mind that the Packet 50 and the Packet 100 are two seperate contests.

7. An entry consists of all the QSL cards needed to verify your entry (50 or 100 depending on which contest you are working on), along with your own signed QSL card and your telephone number. Send your entry to:

   CTM Packet $weepstakes
   Contest Editor
   1704 Sam Drive
   Birmingham, AL 35235

   Please do not send in your entry until you have completed all of the requirements for a particular contest. QSL cards will be returned only if you provide an SASE for them. You will be notified when your entry has been verified by our staff.

8. There is no ending deadline for this contest. The contest will continue

From: Pete WB9FLW

even after all prizes have been awarded. A handsome certificate that you can be proud of will be awarded to all who submit valid entries. Each month, results of the contest, as we have them, will appear in the contest section of CTM.

9. Prizes will be awarded at the next Dayton Hamvention following verification of the entry. In the event that you are unable to attend the Dayton Hamvention, other arrangements will be made. At this time, the prize list is still growing. As it sits right now, this is way the prizes will be distributed. This prize list will be finalized in the January 1986 issue of CTM.

CTM PACKET 50

First Prize: Icom state of the art 2 meter rig (currently the IC-271H with AC supply)

Second Prize: Heathkit HD-4040 Terminal Node Controller

Third Prize: Kantronics Terminal Node Controller

Fourth Prize: GLB PK-1 Terminal Node Controller

Fifth Prize: AEA PK-64 Commodore 64 Terminal Node Controller and your choice of software program

CTM PACKET 100

First Prize: ICOM state of the art HF station (currently the IC-751 dream station with the IC-2KL amplifier and the IC-AT500 antenna tuner)

Second Prize: Kantronics Terminal Node Controller

Third Prize: Heathkit HD-4040 Terminal Node Controller

Finally, we would like to make a minor request from all contest participants. Please drop us a line telling us that you are participating in the contest and from time to time drop us another card or letter letting us know how well you're progressing as well as any comments or suggestions that you may have. Please send these cards and letters to the address mentioned above for entries. As we receive information from you, we will print it here in the contest section for all to see. Good Luck!

-Signed-
Chet Lambert, W4WDR
Editor/Publisher

-Signed-
Jim Griffith, WA5RAX
Packet Editor

-Signed-
Glenn Popiel, WA4FTX
Contest Editor

For additional copies of the contest rules, send a self addresses stamped envelope (SASE) to: CTM Packet $weepstakes, Contest Editor, 1704 Sam Drive, Birmingham, AL 35235.

M 2923 Pete Eaton (WB9FLW,2970) 10/12/85 3:18 PM L:18
KEYS:/I'VE GOT AN UNEASY FEELING ABOUT THIS/
TO: (Group 95), WB9FLW

To: All
From: Pete WB9FLW

Subject: I've got an uneasy feeling about this

The announcement by Computer Trader Magazine of a Packet Contest is one respect is exciting, that of promoting Packet Radio. However in another light I've got an uneasy feeling about it, stated simply:

"Can you have an on-the-air contest, the winner to receive merchandise as a prize?"

A couple of us at the ARRL convention (were the flyers were handed out) had the same question, anybody got an answer?

Oh yes the distribution on this is -OPEN-, I'm sure it will generate some "packets-in-flight"

M 3628 John Gates -NAPRA- (N7BTI,2988) 10/15/85 2:13 AM L:5
KEYS:/SCATTER/6 METER/

Am looking (on behalf of WA7ETE, Steve here in Seattle) for anyone interested in working packet on 6m meteor scatter. Steve wants folks about 1000 miles from Seattle. If you are interested or know of anyone please let me know on DRNET.
Regards, John

M 3128 Skip Hansen (WB6YMH,2964) 10/13/85 12:46 PM L:32
KEYS:/WESTNET GOES EAST/
TO: (Group 95), ARRL

Jeff,

Westnet has finally made it out of the state! Two Digipeaters, WA7HXO-1 and K7WS, have been recently installed on Mount Potosi located southwest of Las Vegas at 6000 feet. They link Los Angeles to Las Vegas thru the AA6TN-1 digipeater in the Big Bear Lakes area, all on 145.01. Extending the path thru existing digipeaters to Sacramento would look like:

K7WS -> AA6TN-1 -> NK6K-1 -> W6AMT-2 -> W6AMT-1 -> WA6RWN -> W6BXN -> K6QIF

The following is forwarded from K7WS:

THE K7WS DIGI WAS INSTALLED ON THE 28TH OF SEPT.
SITE ELEVATION APROX. 6000 FEET ASL
THE EQUIPMENT AT K7WS IS A MOTOROLA METRUM II RUNNING 15 WATTS
ANTENNA 6DB STATION MASTER AND TNC IS A GLB.
NO SPECIAL MODS , RUNNING GLB'S SOFTWARE
WE WOULD LIKE TO GO 220 AND 9600 BAUD HERE IN VEGAS AND HAVE
POTOSI FOR A LINK SITE
PLAN: 145.01 DIGIS ON POTOSI, ANGELS, GOLD MTN (NEAR GOLDFIELD), AND
CHRISTMAS TREE PASS (NEAR LAUGHLIN, NV)
THE WHOLE SOUTHERN 1/2 OF THE STATE OF NV AND PARTS OF UTAH, AZ AND CA
WILL HAVE COVERAGE WITH THESE DIGIS
PLANS TO LINK TO PHOENIX,SALT LAKE AND RENO ARE IN THE WORKS
THERE WERE 6 PACKETEERS IN VEGAS WHEN THE DIGI WENT ON THE AIR...
NAME OF GROUP: NEVADA AMATEUR RADIO ASSOCIATION OF LAS VEGAS, INC.
BY WAYNE SCHENK (K7WS) AND RANDY CURRAN (WA9PPD).

I hope this will be useful information for Gateway. 73's Skip WB6YMH.

M 3194 Phil R. Karn (ka9q,2979) 10/13/85 6:03 PM L:20
KEYS:/PACKET CONTESTING/
TO: (Group 95), ka9q

I see a fatal flaw in the contest rules themselves, ignoring the legality of awarding prizes for amateur operating.

The announcement by Computer Trader Magazine of a Packet Contest is one respect is exciting, that of promoting Packet Radio. However in another light I've got an uneasy feeling about it, stated simply:

"Can you have an on-the-air contest, the winner to receive merchandise as a prize?"

A couple of us at the ARRL convention (were the flyers were handed out) had the same question, anybody got an answer?

Oh yes the distribution on this is -OPEN-, I'm sure it will generate some "packets-in-flight"

M 3628 John Gates -NAPRA- (N7BTT,2988) 10/15/85 2:13 AM L:5
KEYS:/SCATTER/6 METER/

Am looking (on behalf of WA7ETR, Steve here in Seattle) for anyone interested in working packet on 6m meteor scatter. Steve wants folks about 1000 miles from Seattle. If you are interested or know of anyone please let me know on DRNET.
Regards, John

M 3128 Skip Hansen (WB6YMH,2964) 10/13/85 12:46 PM L:32
KEYS:/WESTNET GOES EAST/
TO: (Group 95), ARRL

Jeff,

Westnet has finally made it out of the state! Two Digipeaters, WA7RXO-1 and K7WS, have been recently installed on Mount Potosi located southwest of Las Vegas at 8000 feet. They link Los Angeles to Las Vegas thru the AA6TM-1 digipeater in the Big Bear Lakes area, all on 145.01. Extending the path thru existing digipeaters to Sacramento would look like:

K7WS -> AA6TM-1 -> WB5K-1 -> W6AMT-2 -> W6AMT-1 -> WA6MWM -> W6RXN -> K6OIZ

The following is forwarded from K7WS:

THE K7WS DIGI WAS INSTALLED ON THE 28TH OF SEPT.
SITE ELEVATION APROX. 8000 FEET ASL
THE EQUIPMENT AT K7WS IS A MOTOROLA METRUM II RUNNING 15 WATTS
ANTENNA 6DB STATION MASTER AND TNC IS A GLB.
NO SPECIAL MODS , RUNNING GLB'S SOFTWARE
WE WOULD LIKE TO GO 220 AND 9600 BAUD HERE IN VEGAS AND HAVE
POTOSI FOR A LINK SITE
PLAN: 145.01 DIGIS ON POTOSI, ANGELS, GOLD MTN (NEAR GOLDFIELD), AND
CHRISTMAS TREE PASS (NEAR LAUGHLIN, NV)
THE WHOLE SOUTHERN 1/2 OF THE STATE OF NV AND PARTS OF UTAH, AZ AND CA
WILL HAVE COVERAGE WITH THESE DIGIS
PLANS TO LINK TO PHOENIX,SALT LAKE AND RENO ARE IN THE WORKS
THERE WERE 6 PACKETEERS IN VEGAS WHEN THE DIGI WENT ON THE AIR...
NAME OF GROUP: NEVADA AMATEUR RADIO ASSOCIATION OF LAS VEGAS, INC.
BY WAYNE SCHENK (K7WS) AND RANDY CURRAN (WA9PPD).

I hope this will be useful information for Gateway. 73's Skip WB6YMH.

M 3196 Phil R. Karn (ka9q,2970) 10/13/85 6:03 PM L:20
KEYS:/PACKET CONTESTING/
TO: (Group 95), ka9q

The rules outlaw "store and forward" messages, but they permit the use of digipeaters. What is a digipeater if it isn't a store-and-forward message relay device? The only difference is that it doesn't hold as much at one time, and it only transmits it once. Other than that, they're conceptually the same.

I am tired of short-term "promotions" such as this one which do very little to promote the state of the art of amateur packet radio (as opposed to the "hype" and QRM levels). If somebody's going to sponsor a contest, why not put a little technical challenge into it? How about the first 1000-mile QSO conducted entirely with K9NG modems? How about the first QSO conducted over a network with automatic routing? And so forth.

If you're going to put up with contests, at least they ought to have a constructive purpose.

Phil

M 3237 Lyle Johnson (WA7GXD,2973) 10/13/85 10:25 PM L:7
KEYS:/ONE MORE THOUGHT ON PACKET CONTESTS/
TO: (Group 95), WA7GXD

One more thought for the %$^&$^& packet contest.

How about "no weekend contacts allowed" with a weekend being defined as noon thrusday through noon monday. The friday should be New Zealands and the Monday should be Hawaiis.

Lyle

M 3295 Jack Brindle (WA4FIB,2963) 10/14/85 2:05 AM L:41
KEYS:/SOAPBOX REQUESTS/
TO: (Group 95)

Greetings all Packeteers. I am Jack Brindle, WA4FIB, and have recently joined the ranks of those here on DRNET. I am still stumbling around here, but have managed to find a few things. In way of introduction, my background is communications engineering, both in Hardware and Software. I spent 3-1/2 years working with K4GFG at Motorola in Ft Lauderdale, and recently parted with Paradyne in Largo, Fl where I was in the Network design group. My specialty has become network communications. I am the author of the MacPacket series of programs for the Apple Macintosh.

My current work involves (among other things) the design of a session layer protocol for packeting, which will be unveiled at the Southnet Packetfest in November.

In reviewing the various DRNET conferences, I have found quite a bit of discussion on two of the three layer 3 routing methods, and discussion of TAPR's new NNC. However, I have not found a solid proposal for a Network layer design! In fact, the only "solid" proposal to be found in amateur packet literature is WB4JFI's modified X.25 layer 3 protocol published in the proceedings of the third ARRL Networking conference. Now, before Phil jumps on me for overlooking his TCP/IP proposal, let me note that it is in the form of "what we should do" instead of an actual protocol specification.

Now, I have my own preferences for just how the network should look, but, I believe that those who are arguing about the virtues of the various methods should stop, regroup, and come out with a solid specification detailing the layer 3 header information along with a complete set of implimentation notes so that we can proceed with writing code. There are programmers out now trying to code a layer 3, but with the only spec commonly available being Fox's incomplete spec (among other things missing are details on the implimentation of the very important priority handling proceedures), it is the target of most of these efforts.

There seems to be a lull in DRNET activities, I hope that can be interpreted as network designers writing the specs. If not, think of this

one: We MAY be stuck with an unworkable network layer unless we get off our "pads" and get this info out!

Now I will climb down off my soapbox to hear responses of others. As I said before, I also have some ideas on how a packet network should be designed. Come to the Southnet Packet Conference in November to find out what and how!

73's to all, and I hope we start seeing some protocol specs appearing on the system. (If some are already here, please excuse my tirade and point me in the proper direction).
- Jack B. WA4FIB.

M 3299 Phil R. Karn (ka9q,2979) 10/14/85 2:28 AM L:37
KEYS:/"LEVEL 3"/
TO: WA4FIB, (Group 95)

Jack, the reason I did not put a complete protocol specification into my paper for the 4th proceedings is very simple: they have already been published in other forums, and since they are entirely suitable as-is for our needs I felt it necessary to include them only by reference. I have shipped out quite a few paper and electronic copies of the ARPA documents specifying IP, TCP and ICMP, and I have also recently received a paper copy of MIL-STD-1778, the "DoD rewrite" of the TCP spec. I will be shipping a copy of the latter to the ARRL Digital Committee (as I have with all my other documents) and I can sustain a VERY LIMITED capability to satisfy direct requests (SASEs are greatly appreciated, as these specs are a few hundred pages each..

The only specification which was necessary is the one describing the interface between IP (as layer 3) and AX.25 (as layer 2); this was discussed in detail in the paper and summarized in the appendix. As of yet, however, I have received no comments from the Committee or anyone active in amateur packet radio, although I have received a few helpful suggestions from amateurs within the ARPA community who seem to have a latent interest in packet radio.

This seems to be a good time to mention that I am nearing completion of a "C"-language implementation of the TCP protocol for general purpose amateur packet radio use. Since one of the concerns about this protocol has been its complexity and implementation size (and whether it would fit on a TNC), I thought it would be interesting to give the current object-code size of all the modules making up TCP: 5K. That's right, FIVE kilobytes. That's all. If we are to do networking RIGHT (i.e., by avoiding "transport level bridges", this protocol belongs in each and every TNC. The "network controller" boxes can then concentrate on the jobs of packet switching and routing, which are sufficient challenges in themselves without having to support some kind of "user interface".

Further discussion on this subject should be moved to the "amateur radio level 3" conference.

73, Phil Karn, KA9Q

C2987 CC199 Skip Hansen (WB6YMH,2964) 10/13/85 10:00 PM L:27
KEYS:/MULTICONNECT TRANSPARENCY/

Howie,

I just finished wirewrapping a 2764/27128 programmer and it works! Now I can stay current without relying on the mail. I tested the programmer by burning the 1.1.1f code, just installed it and it fired right up.

I am very interested in going to multi-user on my host with two R.F. ports and multi-connect on each port. So far I haven't started writting any code, but have been giving the problem a lot of thought.

The only problem area I see with the current configuration is transparency. I currently run my TNC in transparent mode to allow binary file transfers (I have had a binary file transfer program on line for over a year, but only two stations have used it in binary mode so far). I believe binary file transfer capability will become more important as time goes on. I would like to suggest that the TNC software be modified to handle stream switch characters received in incomming packets a bit differently than the present implementation. I would like to see the stream switch character simply doubled when it is sent over the serial port. This would preclude using "A","B","C",or "D" as the stream switch, but I think that would be an acceptable limitation! This would allow transparency with very simple software on the host end. It same technique could be used for sending characters to the TNC as well. If user=1 then all of this would be disabled of course. What do you think ?

73's Skip WB6YMH

C2987 CC200 Howard Goldstein (N2WX,2987) 10/14/85 7:47 PM L:38
KEYS:/16K RAM TNC 2/1.1.1/CKSUM $2F/+READ THIS ITEM/

To: Beta
Fm: Howie
Re: 16K RAM version 1.1.1g, ten connection, release description
Dist: Closed, for now

The .HEX of the release described hr can be had with +READ C2987 CC200.

TNC 2 software for 16K RAM is done. Buffer enlargement is one of the things addressed. Here are some other changes:

o - Up to TEN connections/links are allowed

o - A clock adjustment is installed. The associated command is CLKadj nnn, where 0=<nn=<65535. The adjustment effects the time by subtracting 9.1667ms every 100nn msec. Ex:
only WHEN CLKadj != 0,

$$\text{speed change of clock, in \%} = 100 - 9.16667 * \frac{1}{\text{CLKadj}}$$

o - *** DISCONNECT messages are stamped according to CONstamp

o - USER command is changed to USERS, and

o - The USERS parameter has new significance:
USERS = 0 allows incoming connections on any free stream
USERS = 1 allows incoming connections on stream A only
USERS = 2 " " " " " " on streams A & B
USERS = 3 " " " " " " on streams A,B, & C
etc...

o - A bug where the output stream didn't change on ^C from CONV mode on a different stream has been fixed.

73 Howie

C2987 CC202 Howard Goldstein (N2WX,2987) 10/14/85 7:56 PM L:8
KEYS:/MULTI CONNECT TRANSPARANCY/
A: 199

Skip - That's one of the ways to do it, but it still wouldn't be failsafe when the remote sends your STReamsw a number of times in a row. (Escaping the escape which is escaping the next escape which is....)

I agree that converting them all to "$" does leave something to be desired as far as transparancy/finesse.

C2973 CC7 Phil R. Karn (ka9q,2979) 10/20/85 8:27 PM L:343
KEYS:/W3HCF TCP/IP PROGRESS REPORT/

>From @DCN7.ARPA:mills@dcn6.arpa Fri Oct 18 15:38:56 1985
Relay-Version: version B 2.10.2 9/18/84; site petrus.UUCP
Posting-Version: version B 2.10.1 6/24/83; site mit-eddie.UUCP
Path:
petrus!bellcore!decvax!ucbvax!ucdavis!lll-crg!seismo!harvard!think!mit-eddie!@DCN7.ARPA:mills@dcn6.arpa
From: @DCN7.ARPA:mills@dcn6.arpa
Newsgroups: net.ham-radio.packet
Subject: AX.25 and IP/TCP: a status report (absurdly long!)
Message-ID: <131@mit-eddie.UUCP>
Date: 18 Oct 85 19:38:56 GMT
Date-Received: 20 Oct 85 18:52:34 GMT
Sender: daemon@mit-eddi.UUCP
Organization: MIT, Cambridge, MA
Lines: 326

From: mills@dcn6.arpa
Folks,

Following is a longish report on current status and plans of my ongoing project to bring up a demonstration of the DoD Internet protocol suite operating at the network layer above AX.25. I first tried to hack the original TAPR-1 firmware to do this in both connection and connectionless modes, but gave up due to ornery problems in transparency and flow control. I next tried the WA8DED firmware and found some joy, pleasantries of which are reported below. The report is rather technical and intended for those with some protocol background and implementation experience, expecially in the areas of connectionless (datagram-oriented) networks and gateways.

Overview

I rebuilt what has become a monsterous driver for the LSI-11 "fuzzball" system, which is an old grizzly used for network protocol development, testing and performance evaluation and supports just about every protocol sung in the IP/TCP protocol suite. The LSI-11/73 I used for the present purpose (DCN6.ARPA) belongs to me personally and is located in my home along with 30 megabytes of disk, a swamp of radios, towers, a leased digital circuit to the office (hence the Internet) and related mud creatures.

The driver supports both the ordinary connection-mode, stream-ASCII operation commonly used to access BB stations, etc., as well as both connection and connectionless modes for Internet Protocol (IP) datagrams and its client Transmission Control Protocol (TCP), upon which application protocols like the TELNET virtual-terminal, FTP file-transfer and SMTP mail-transfer protocols are built. The station complement, consisting of TAPR-1, fuzzball and VHF transceiver, is organized as a full-function Internet gateway between subnet 128.4.1, representing the radio channel, and DCNET (128.4), which is itself gatewayed to the Internet via the ARPANET. It should be noted that no traffic capability between the radio channel and the ARPANET is expressed or implied in this document and that all traffic between the radio channel and any other nets reachable via nongovernment paths (Ford, U Michigan and U Maryland at present) is experimental and strictly noncommercial in nature.

The WA8DED Firmware

The WA8DED firmware ex box provides one connectionless-mode channel and five connection-mode channels, which can also be operated in connectionless mode. The TAPR is operated in host mode; that is, the fuzzball has to poll for everything, which it does at one-second intervals on all five channels. This handles the flow-control problem from the TAPR to the fuzzball. Since my TAPR is fully populated with memory (about 700 buffers) and TCP manages end-end flows anyway, I didn't bother with flow control from the fuzzball to the TAPR.

Host messages to the TAPR consist of the sequence

<channel number> <control code> <count> <text> ,

where <channel number> is an octet in the range 0-4, <control code> is an octet with value 0 for data and 1 for command, <count> is an octet representing the number of octets in <text> reduced by one and <text> is a string of up to 256 octets with no restriction on code combinations. Commands are in ASCII and are interpreted the same way as in ordinary (i.e. non-host) mode, except that the terminating <CR> is not included. No more than one command can be sent in a message and commands may not be split between messages. In the case of IP datagrams, the raw datagram itself is simply encapsulated in the <text> field as-is.

TAPR messages to the host have the same format as above, except that <control code> is an octet with values in the range 0-7, indicating success/failure, as well as status information, headers and data.

0normal response to command (no <count> or <text> field)
1normal response to command
2error response to command
3link status information
5monitor header
6monitor data
7user data

In all except codes 0,6-7 the <text> field is <NUL>-terminated. Codes 0-2 occur only in response to commands, while the remainder can occur spontaneously (but only in response to a poll, of course). Codes 5-6 apparently occur only on channel 0, while code 7 occurs only on channels 1-4. This creates problems in connectionless mode as described later. In the case of code 5 the <text> field consists of an internally formatted representation of the AX.25 header, which is followed immediately by the data (if present) as a code 6.

Upon initializatin the driver sends the sequence

<NAK><ESC>JHOST 1<CR> ,

which toggles the TAPR into host mode, if not already there, and is grumpily ignored if it is. The polling sequence, which is sent every second, consists of the five commands

<0><1><0>G <1><1><0>G <2><1><0>G <3><1><0>G <4><1><0>G ,

which empties the TAPR queues at a rate limited to one message per second on each channel. When lots of things are going on, such as when monitoring all channel traffic and managing a connection or two, the TAPR queues start to back up. The polling strategy and rate matching can obviously be improved.

Connection Management

The WA8DED firmware assigns incoming connection requests serially starting at channel 1 and continuing to channel 4 or a limit which can be set by a command. Completion of the handshake sequence results in a link status (code 3) reply on the assigned channel. Outgoing connection requests can be made on any channel as specified by the connect "C" command. In the case of channels 1-4, the request results in the AX.25 connection-setup procedure, while in the case of channel 0, the request specifies the route for subsequent connectionless (UI) frames.

If a data message is sent on channel 0 or on any other channel not preceded by a connect command, the data is sent as a UI frame. Connection data (I frames) appear as code-7 replies on the assigned channel and can be interleaved with data from other channels. Connectionless data (UI frames) can apparently be

received only as monitoring data when enabled by the monitor "M" command and are preceded by a code-5 message indicating the route and protocol id. The protocol id can be changed by a command, but only on a global basis.

An AYT (ARPANETese for "are you there" aka "keep-alive") consisting of the RR supervisory message, is sent every two minutes on every active connection. If the max-retransmission counter is exceeded the TAPR attempts to reset the connection while holding on to the channel queues; however, if that fails the connection is aborted with an appropriate link status message. This can result in delays up to several minutes before the link-level protocol gives up and obviously will result in congestive collapse in retransmissions with conventional transport-level implementations.

The W3HCF Fuzzware

The TAPR driver constructed for the fuzzball can operate with both ASCII streams or IP datagrams in either connection or connectionless modes. The driver has an interface to the stream-oriented terminal I/O, so that local terminals and remote network (TELNET) terminals can connect via the driver and TAPR to AX.25 stations in the ordinary way. Using this interface, if the first character in a line terminated by <CR><LF> is <ESC>, the line is interpreted as a command, rather than data. If the first character of the command itself is a digit, the selected channel is changed accordingly and remains in effect. In addition, a rather ad-hoc set of editing rules for <CR> and <LF> had to be implemented for compatibility with local BB stations, so this interface is necessarily code sensitive.

In the case of IP datagrams, a full-bore IP-address/callsign dynamic mapping cache was implemented using two tables, the Channel Table and the Route Table. The Channel Table latches the current status (state and time-to-live (TTL) counter), last non-data message received and callsign. The callsign is used as the local-network address in the IP model. Following is an example (slightly edited) display captured during normal operation:

```
CID  CallsignStatusMessage
---------------------------------------------------------------------------
0    W3HCF   0  05 fm W3IWI to KA3DBK via WB4JFI-5* ctl I17 pid F0
1    W4HCP    0  03 (1) DISCONNECTED fm W4HCP via WB4JFI-5
2             0  00
3    W3HCF   2  1200 @PF0
4*            0  01 CHANNEL NOT CONNECTED
```

The CID field indicates the channel number, with the asterisk indicating the currently selected one, while the Status field is coded as first the state and then the TTL. The intent is that, if the TTL counts down to zero the channel is closed. The reason for this will be described momentarily. The Message field begins with the <code> field (codes 1-5 only) of the last message from the TAPR followed by the <text> field as-is. as received. A code of zero indicates the last message was a command sent to the TAPR. In the above, channel 2 has never been used.

The Route Table is maintained as a cache using mechanisms similar to the Address Resolution Protocol (ARP) used with Ethernets. It consists of an associative array of two-field entries, one field containing the IP address and the other the associated digipeater route. The table is presently handcrafted; however, plans are to dynamically manage the entries as the result of monitored channel activity, perhaps augmented by ARP-stype broadcasts. Obviously, this is where the fun lies in the protocol-development area. Following is an example display of the current version:

```
IP addressRoute
------------------------------
[128.4.1.1]W3HCF WB4JFI-5
[128.4.1.2]W4HCP WB4JFI-5
[128.4.1.3]WD5DBC WB4JFI-5
[0.0.0.0]W3HCF WB4JFI-5
```

Note that the contents of the Route field can be used as-is as the argument in a TAPR connect command. In this particular case, the local radio channel is configured as subnet 128.4.1 of net 128.4, which is the swamp in which our own fuzzballs sleaze. Also note the default [0.0.0.0], which matches all other IP addresses, used for loopback and testing.

Connection Mode

The whole contraption can operate in either connection or connectionless mode. In connection mode it operates like this. Assume no channels are connected and a datagram originates at the fuzzball or arrives from another link destined for 128.4.1.2, previously determined to belong to W4HCP and to be reachable via the WB4JFI-5 digipeater. The Route Table is searched for 128.4.1.2 and yields the route string W4HCP WB4JFI-5. Next the Channel Table is searched for an active channel matching the destination callsign W4HCP in that string, keeping track of the highest-numbered inactive channel in the process and avoiding the monitor channel (0) and the currently selected ASCII channel (4 in the above display). If the match succeeds, the datagram is immediately sent on the associated channel; if not, a connection command specifying the route string is sent on the higest-numbered inactive channel and that channel is marked active. The datagram starting all this is sent following the command and is queued in the TAPR. Note that all datagrams are sent with protocol id CC (hex), which is assigned to the IP network layer in AX.25.

Datagrams arriving from the TAPR on an active channel are simply tossed into the normal fuzzball routing algortihm just as if they arrived from other links. A transmitted or received datagram causes the TTL field in the Channel Table entry to be reset to its maximum. The TTL fields of all active channels are decremented once in a while and, if one of them counts down to zero, a disconnect command is sent to the channel, allowing it to be used for possibly other destinations. Note that the AYT mechanism mentioned above is also operative and would normally catch a broken station before the TTL mechanism did. The TTL mechanism is intended primarily as a garbage collector when channels lie dormant for unreasonable periods.

Presumably as the result of monitoring activity, amendments are continuously being made to the Route Table. If a connection breaks with datagrams queued, the queue will be discarded and a new connection attempted according to the latest information in the Route Table. The lost datagrams will be recovered through the ordinary retransmission mechanisms of the client transport protocol. Obviously, the key to the success of this scenario lies in the ability to effectively collect the route information and maintain the Route Table. Further development is anticipated in this area.

Connectionless Mode

In connectionless mode all IP datagrams are sent and received on channel 0. This requires considerable heroics in separating the desired IP traffic from other traffic on the channel, not to mention monitoring headers and data. A datagram originating at the fuzzball or arriving from another link initiates the same operations as in the connection mode, as described above, except that channel 0 is used and it is never marked active. This implies some overhead, since a connection command is always sent preceeding every datagram. It is easy to avoid this by the simple expedient of saving the last argument and suppressing this if the argument is unchanged.

At present the only mechanism to capture IP datagrams from the radio channel is by turning on UI-frame monitoring (Ethernetters know this as promiscuous mode) and carefully filtering IP datagrams from the general slosh. The Thursday-night implementation (which should be obsolete by the time you read this) is to toss everything at the IP-header validation routine, which verifies correct IP header format and checksum and with high probability chucks everything but IP datagrams on the floor.

Initial Experiments

The driver is built so that ordinary monitoring headers and data are sent to the system log file, which allows channel activity to be observed in real time, as well as captured for later analysis. During initial testing it was obvious that a fair number of other protocol implementations active in the local community are defective, inefficient or worse. A report on that will be presented at another time. Specific IP tests were conducted in connection and connectionless mode via the WB4JFI-5 digipeater, which is on a high tower and has a very strong signal in the W3HCF local area. It can then be assumed that almost all packet losses are in the direction from W3HCF to WB4JFI-5 and those due to collisions aggravated by the hidden-terminal problem.

Initial experiments used virtual-terminal (TELNET), file-transfer (FTP) and raw-datagram (ICMP Echo) protocols. In general, the tests were successful and demonstrated that the Internet protocols worked handily via the lashup. However, several problems were immediately apparent, among them:

1. The speed of the fuzzball-TAPR connecting line is only 1200 bps, so that nontrivial queuing delays due monitoring and status information place exceptional strain on the TCP retransmission-delay estimation algorithm. The result is an unneccessarily high incidence of TCP retransmissions. Obviously, the cure for this is to hop up the speed, which will have been done before you read this.

2. In connection mode when times get tough the TAPR tries heroically to get the traffic through, which may involve several minutes of retransmissions, resets, etc., during which TCP is retransmitting and possibly abandoning the connection. Even if the traffic does get through, the result is gobs of useless duplicates and end-end ACKs which clutter up the channel. The cure for this is to reduce the max-retransmission limits and short-circuit the reset procedure.

3. I have observed that, on average, between one packet in four and one packet in ten is lost on the local radio channel. This is a high loss rate even for TCP, which is designed for a more lossy environment even than HDLC, let alone AX.25. Casual inspection of the monitoring log shows inspired braindamage on the part of some local AX.25 implementations using excessively large window sizes, which lead to excessive retransmissions at such high loss rates. This problem has nothing intrinsically to do with IP or TCP and can be cured by judicious intervention by the Protocol Police.

Conclusions and Plans

The "Nagle Algorithm" in conjunction with other refinements developed for the fuzzball TCP implementation was designed specifically to minimize spurious packet traffic and in principal could result in much improved performance. While the packetization and estimation algorithms designed for the present generation of TCP implementations have been designed for an extremely wide range in delays, speeds and loss rates, they have not been optimized for the extreme regime represented by a 1200-bps packet channel with loss rates in the neighborhood of one in four. The fuzzball TCP implementation, which has been designed and heavily instrumented for protocol development, is a useful tool to explore these issues.

>From a purely engineering perspective the use of ARQ techniques in such a regime is flawed at best and should in any case be augmented by some form of FEC coding. In addition, the use of compression techniques (e.g. the SLIP protocol) should be explored as a mechanism to reduce packet-header overheads. Exploration of these issues may lead to revised packet formats and checksumming procedures. One of the first results expected to come out of this investigation is a requirement for separate checksums on the link-level header (AX.25 address, route and control bits) and data portions of the frame. The expectation is that residual "errors" alleged at the AX.25 level would be corrected by its client while nevertheless preserving the integrity of the routing and framing functions.

Probably the most effective realization of connectionless mode in the AX.25 paradigm would involve implementation of at least the IP-datagram support in the TAPR or equivalent. This is not really hard at all and amounts to adding a feature, keyed on the presence of CC in the protocol-id field together with a specific destination callsign, to indicate an IP message to the client host, which should be delivered with AX.25 header intact in order to support dynamic route cacheing as discussed previously. It should be possible to specify a list of such callsigns, which could then be used to implement a multicast/broadcast capability similar to Ethernets. This feature should be completely independent of the monitoring facility. Finally, it will be important in future to implement a filtering mechanism, based on the protocol-id field, which allows the casual user to avoid screen clutter due to unprintable code combinations while monitoring non-vanilla protocols.

Dave

C2973 CC8 Jack Brindle (WA4FIB,2963) 10/21/85 12:11 AM L:42
KEYS:/DATAGRAMS TCP IP/

Phil; After reading your lengthy memo, I have some comments. After noting the new report here, I think I will reserve many of my comments until afterr I can read it. However, I would like to ask a few questions of you.

In the TCP/IP environment, are you envisioning several levels of network, of the local, regional, national levels? Otherwise, Does each tnc require an International Electronic Callbook to be able to route packets?

I have some problems requiring each tnc to have layer 3 implimented. To put it another way, how are you going to explain to literally thousands of Kantronics owners that they can no longer participate in packet radio because their tnc does not have (or for that matter, has no chance of) the AX.25 network layer protocol???

After blasting you, I should offer some alternative. Well, it will be forthcoming, Real Soon Now... I would like to get a chance to further study your proposal before really commenting on it. I am curious, though, as to the exact nature of your current project and its relation to the memo you loaded on the system (apparently off ARPAnet). You previously mentioned your implimentation of IP. Are you also doing TCP as would be required in a network? (If not, then what good is IP?) And, does the code run atop your layer 2 code? I would be very interested in obtaining a copy of your C source (to run on a Mac, of course - I have so far resisted obtaining an 820 although I am beginning to see it coming...). Whether or not it will be the "chosen" protocol is rather immaterial now, since it is the only game. It does give us the opportunity to play with networking, however, to demonstrate to the "masses" that we are actually doing something.

On another subject. I am looking at doing a 68000 based NCC. The current design has 3 8530 SCCs (1 polled/interrupt, the other 2 DMA), a 6522 VIA (or 2) and 20 28 pin sockets for byte wide memory chips. Four of these chips would be located at low memory, and would be accessable at all times by the cpu, the remaining 16 would probably be ram accessable both by the cpu and dma controller. The dma controller has not been chosen at this time, but I don't want to use the Motorola dual channel chip. I am also looking at a time of day chip of some sort, probably serial. This looks to be a rather powerful system, and is not necessarily meant to compete with the TAPR NNC, but rather to give us a more powerful controller for higher speed use. If you have any comments or suggestions, pass them along. My feeling on this board is to allow the user to "mix and match" the peripherals for his needs.

More later, after I read the long message. I also need to figure out this system more, like just how to respond to someone who sends me a (an) interactive message. Maybe I need the eies manual after all!

73, Jack B.

C2973 CC9 Phil R. Karn (ka9q,2979) 10/21/85 6:41 PM L:69
KEYS:/NETWORK PLANNING/

Jack,

Dave Mills' work is complementary to my own. He is trying to gain some early experience with running the Internet protocols on top of AX.25, using off-the-shelf components at each layer (existing TNCs and Internet hosts). My job has been to design new implementations for AX.25/IP/TCP that are designed to fit together more cleanly on a small, standalone machine. I wrote the AX.25 code with the necessary software "hooks" to the bottom end of an IP layer, and if it can all fit on a single machine (and it now appears that it can) then the interfaces (especially the one AX.25 presents to IP) will be much cleaner than the ones Dave is struggling with.

The structure of the network (regional/local/etc) is a question of addressing and routing, not the protocols. In any event, my second conference paper (in the 4th Proceedings) gives my thoughts on this subject. Rather than repeat the whole paper here, I'll just summarize my main point.

For a network of reasonable size, we need a hierarchical addressing and routing scheme that also permits a somewhat arbitrary topology (e.g., ad-hoc satellite links). I am proposing an addressing scheme that is a hybrid between the complete topological flexibility of flat addressing (where every station has a complete list of every other station, so that topology changes don't require readdressing) and a hierarchical scheme where (most) addresses depend on the network topology in such a way that routing table space is conserved. I would welcome any comments on this paper.

The end-user nodes (the "hosts" that speak TCP) need not implement full-blown routing mechanisms. A common strategy on small machines (e.g., IBM PC's) that speak TCP is to send all traffic destined outside the local area to a "default gateway". The gateways are of course full-blown IP packet switches, and contain complete routing tables. It is possible for there to be several gateways out of one local network, so if the gateway is given traffic that is more optimally routed through another gateway, it may inform the user host of this with an ICMP Redirect message. In this way the user host may build up a small routing table of its own, but it need contain only those sites with which it actually communicates, and only if these sites are best reached through a gateway other than the default.

I understand your concern about thousands of existing Kantronics TNCs. I cringe every time yet another manufacturer comes out with a box just barely able to squeak by at 1200 baud with AX.25. Packet radio is a rapidly evolving field, and it is clearly shortsighted in the extreme to design units that barely operate satisfactorily with current standards, to say nothing about leaving room for future developments. In any event, it isn't totally clear yet that TCP *won't* fit on such a TNC, given the reasonably small size of my implementation and the exponentially increasing capacity of EPROMs. It is always possible to put TCP in another box which uses the TNC as just a link controller, in the same way that people plug smart terminals into simple modems. One of the things I hope will come out of Dave's work is a set of recommendations for level 2 TNC designers on the interface it should present to an external IP client (as opposed to a dumb terminal) over the asynchronous port. This should make things easier should the TNC manufacturers decide not to support the whole package in one box.

Our biggest mistake in amateur packet radio is that we have been building the networks from the bottom up, instead of the top down. For example, we have formally adopted only a link level protocol, but because every network must have a transport and network protocol we are in fact using AX.25 for those functions. Changing over to "the right way" can be pretty painful, especially when amateur packet radio has been growing the way it is (way too fast for its own good, I might add.) In retrospect, we should have first established a standard TRANSPORT layer protocol (i.e., TCP), and then worked downward as we felt the need for more sophisticated network layer services (e.g., automatic routing) and link layer services (e.g., hop-by-hop acknowledgements). There was in fact a proposal from KA6M to do just this way back in 1981, but unfortunately it was run over by the AX.25 steamroller.

Phil

C2973 CC10 Phil R. Karn (ka9q,2979) 10/22/85 12:11 AM L:43
KEYS:/YOU THINK I GET EXCITED?/

From brian@SDCSVAX.ARPA Thu Oct 17 14:48:16 1985
From: brian@sdcsvax.arpa (Brian Kantor)
To: karn@mouton.arpa
Subject: you'll enjoy this
Status: R

Wish he had a ham license...

> From fair@ucbarpa.Berkeley.EDU Thu Oct 17 10:37:14 1985
> Received: by sdcsvax.ARPA (5.28/4.41)
> id AA00619; Thu, 17 Oct 85 10:34:48 PDT hops=0
> Received: by UCB-VAX (5.28/5.13)
> id AA13164; Thu, 17 Oct 85 04:51:12 PDT
> Received: by ucbarpa (5.28/5.12)
> id AA05983; Thu, 17 Oct 85 04:51:05 PDT
> Date: Thu, 17 Oct 85 04:51:05 PDT
> From: fair@ucbarpa.Berkeley.EDU (Erik E. Fair)
> Message-Id: <8510171151.AA05983@ucbarpa>
> To: brian@sdcsvax.arpa
> Subject: Re: IP/TCP bumps and grinds
>
> Actually, I do know that IP is usually encapsulated in something. Even in Rick Adam's SLIP...
>
> As for ISO, fuck 'em. They haven't produced ANYTHING real. Read Padlipsky's book (or if you're really cheap, read RFC871-875, since they comprise the majority of his book, modulo some very interesting, nd irreverent comments).
>
> The point is that X.25 is a crock of shit, and IP is the right way to go. It is trivial to provide an attached network processor, (like Excelan, CMC, and a bunch of other board manufacturers do for the IP/Ethernet combination) in most of the existing ones, the only thing that the host must do is data demultiplexing (i.e. deliver data to the right processes), because all the other details are handled by the board. Of course, the boards also have to allow raw access to the IP module, and to the network itself, if it is going to provide full functionality...
>
> It would (as I said before) be a damn shame to see packet radio go the wrong way.
>
> Erik E. Fair ucbvax!fair fair@ucbarpa.BERKELEY.EDU

C2973 CC11 Howard Goldstein (N2WX,2987) 10/22/85 8:43 AM L:4
KEYS:/ACIDIC/SURPRIZE/
A: 10

Phil - such vitriole from Mr. Fair! Boy won't he (and all the disbelievers) be shocked when TNC 2's are delivered with **AX.25 Network Layer** installed and working!! 73 Howie

C2973 CC12 Phil R. Karn (ka9q,2979) 10/22/85 1:27 PM L:3
KEYS:/SURPRIZE/

Howie, would you be so kind as to tell us what "AX.25 Network Layer" IS?

Phil

C2973 CC13 Howard Goldstein (N2WX,2987) 10/22/85 2:42 PM L:2

KEYS:/SURPRIZE/NOT THE PLUMBING KIND/
A: 12

Phil - AX.25 network layer is X.25 level 3 sans PVCs. 73 Howie

C2973 CC14 Thomas A. Moulton (W2VY,995) 10/22/85 3:54 PM L:24
KEYS:/NETWORK PLANNING/YOU CAN BURN ALL THE PROMS/
A: 9

Phil,

If your intents get out to the people who have made packet radio so popular (aka Kantronics, Heath, AEA, etc)
We will be FUCKED just like you said back a couple of comments, but instead of it just being the CCITT jerks, it will also be the TCP/IP jerks and the rest of packet radio, and then we might as well go back to RTTY

I think you are saying that we MUST REQUIRE each user to buy "my" box frankly I think our networks should support ALL the people who have these old boxes, hell I'd like to see people using the Vancouver and Ashby boards!

To turn around and tell everyone that they have to sell their WORTHLESS TNC 1's and buy the new TCP 1 will KILL packet radio, that would be a little like telling us we had to use V.22 instead of BELL 212

What we see as a reasonable approach is to ALLOW users to have level 3 PADs but also support level 2 users by letting them connect to the switch, and being prompted much like the PADs the PDNs use (Telenet/Uninet) given time the older boxes will fade away, (like AM operation has) but we shouldn't just turn our backs on them, I think that I would feel this way, even if I did like TCP/IP!!

73, Tom

C2973 CC15 Phil R. Karn (ka9q,2979) 10/22/85 7:15 PM L:8
KEYS:/AX.25 LEVEL 3/

Howie,
I wonder if you are aware that X.25 is an INTERFACE protocol, not a NETWORK protocol? It describes only the means by which a customer gets into the network from outside; it says nothing about how the network functions internally. I think you probably mean X.75 instead of X.25, but even there issues such as routing are left completely unspecified.

Phil

C2973 CC16 Phil R. Karn (ka9q,2979) 10/22/85 7:31 PM L:33
KEYS:/WE BURN PROMS ALL THE TIME/

Tom,
I never said that we MUST REQUIRE each user to buy "my box". What I said was that we must require each user to speak a common transport protocol. I don't care what hardware they run as long as it can speak the proper protocol. As I think I've shown with my TCP implementation, it is small enough to completely dispel the old objections about code size.

Perhaps I have been a little too strong in my condemnation of Protocol Conversion Gateways. It is true that one can lash them together if supporting unmodified TNCs is absolutely necessary. However, it is very difficult or impossible to obtain the full benefits of the Internet style protocols by doing so. You can get "remote login" capabilities pretty easily, just as I can by dialing into my UNIX host from home with a dumb terminal without having to speak IP across the dialup line. However, that's about all you can do with a "vanilla" phone line, and if you've got something more powerful at home than a dumb terminal.

pretty soon you start wishing you could get direct access to the underlying network services.

If you want to build such a "gateway", be my guest; all I'm saying is such a device must not be thought of as a permanent part of the network but rather a temporary stopgap to ease the conversion to true end-to-end transport protocol implementations. I maintain that this is true REGARDLESS of the specific transport protocol that is selected, and I think that even Gordon agrees with me on this point!

My own priority, however, is to build a solid foundation for our network FIRST, before we worry about building conversion stopgaps. If we always placed backward compatibility ahead of everything else, packet radio would be using 5-level coding at 45.45 baud!

Phil

C2973 CC17 Thomas A. Moulton (W2VY,995) 10/22/85 8:55 PM L:20
KEYS:/I LIKE TO TALK TO DX STATIONS/THEREFORE DON'T LIKE US DOD TCP/IP/
A: 16

I fully agree that the transport can be anything, my previous comment was mostly fueled from you comment on cringing when another vendor decides to promote packet radio and put yet another box on the market.

you keep screeming that we should use this or that and saying that X.25 level 3 is this or that, THIS HAS NOTHING TO DO WITH END-TO-END, just the links to get there, I am working on things, we will include routing, using standard addressing which will be based on good old area codes which we all have maps for, just open your local telephone book.

you said you want to get a solid foundation before you worry about the rest
we agree, but why are you doing the transport layer now?
hadn't we better get the end-point to end-point network up first?
that's what x.25 level 3 gives you, then we will write session and then TP 4
on top of it all, I still want to talk to our european friends without having
to go via a gateway, because information gets lost in gateways
and we know how the PTT's feel about protocols
(we're lucky they agree to international standards)

i don't know why i keep beating the dead horse

C2973 CC18 Jack Brindle (WA4FIB,2963) 10/23/85 1:31 AM L:81
KEYS:/THIS RING HAS NO CORNERS.../

Tom; Phil is working on the Transport layer implimentation with good reason. Since we currently have a point-to-point system, we have no need for layers 3 & 4. The transport functions are essentially handled by the link layer. The checking provided for by L4 is being done by the AX.25 L2 implimentation. And, essentially the link layer is handling what the network layer will be doing in the future. Phil is right in that we have a sort of datagram system within layer 2 to provide the relay functions. Since no connections are made a L2 except for end-to-end, it falls short of true networking in either the VC or datagram sense. By the way, one thing to remember is than in X.25, there are only two addresses used at the link layer! Our transmission medium required a major departure from X.25 on this point.

When layer 3 is introduced, we will need a transport protocol to maintain data integrity. The transport layer is more important to the datagram approach because of its use of multiple routes for packet delivery, and thus its tendency to deliver packets out of sequence. Both the Virtual Call and Permanent Virtual Circuit approaches require a good Transport protocol to assure data integrity. The protocol need not be as rigid as that used for datagrams (it would be nice, though). One point that has escaped many people is that packets going through a network do not

necessarily go in sequence, but may become seperated by other user's packets. This may happen for several reasons such as buffer allocation overflows or simply from higher priority traffic. This one point will kill Phil's method of breaking datagram packets into AX.25 sized packets without proper sequencing (again remember that the layer 2 sequences mean nothing at layer 3!, or between the next two nodes in the network).

The point is that we do need a layer 4 protocol, and should be working on an implimentation coincident with the layer 3 work. My initial file transfer work involved layer 7,5 and 4 protocols until I realized that I needed checkpointing in layer 5 (yes, there really is a good use for the session layer). My current approach, and the one I am proposing, uses a session layer protocol to provide checkpointed, and thus error-detected, transfers, not only for files but also for any other type of QSO. The MacPacket implimentation will allow at least 8 concurrent sessions! The specifics of the file transport are handled in the Application layer. This will allow disimilar (sp?) machines to transfer files. They may not recognize the specific formats (A PC would throw up on MacBinary files), but at least they could send and receive them so that the files can be reconstructed on a similar machine in the same form as they were sent.

Note that I can get away with only layers 2, 5 and 7 because of the functions each handles. If I were to allow multiple connects with several stations, layer 4 would be required to seperate the incoming packets and provide the other services the transport layer performs (resequencing, error detection, class of service, etc.).

One question I have for you and the others working on the VC approach involves the handling of priorities. Howie and I have discussed this, and have noted that it is one of the shortcomings of the Tery Fox L3 proposal. Because of the nature of data that we will be transferring, we will need to impliment priority in both layers 4 (through service classes) and layer 3 (through proper queue management to layer 2). The place to put the priority info is in the LCI (I suspect that few packeteers understand the significance of LCIs. For example, a packet coming in from one node and the same packet outbound to the next probably will have completely different LCIs!). My recommendation would be to use the top two bits of the LCI to provide a 4 layer prioritization scheme, that is probably enough for our use. We will need to set up rules for service classes and thus priority so that file transfers receive a lower priority than keyboard QSOs, which are lower than Voice QSOs.

We have a major problem with current TNCs (solved in the WA8DED code) due to their user interface. The fact that the tnc does not report the delivery of a pack (known from the received ack) hurts the upper layers ability to perform error detection (I don't believe DED solves this problem either). We have proven that computers do better with numeric interfaces than human type verbose interfaces. This has hindered my work in developing MacPacket, and has caused many upper level designers to wish for something better from the next TNC software release. The approach the that Ron, WA8DED has taken is interesting in that is resembles the IBM approach of poll/response. That has proven to be quite reliable in practice, and seems to also work well in the TNC1 environment.

Anyway, It seems that we all have a lot of things in common. I believe that Phil has some very valid points, and although I don't believe that datagrams are the best approach (until Phil can prove it to me), I also see that VCs have major problems (also remember, I support the use of PVCs with VC used only to connect a new node to the network on a temporary basis - until the network fully integrates the new node). I am quite willing to work with all those implimenting network software in whatever way I can - even to code it on the Macintosh (please, a C source first). This should be interesting, I can indeed see that the folks have sort of "shut up" while they prepare to "put up". Whatever emerges should be quite interesting.

More later. 73, Jack B.

C2973 CC19 Lyle Johnson (WA7GXD,2973) 10/23/85 1:36 AM L:13
KEYS:/68K NNC/
A: 8

Jack,

.

The 68k NNC sounds interesting. Does it seem reasonable to drive this thing with a DMA'd SCSI bus and let the 64180-style board handle the level two stuff as a front-end processor?

.

I am sure the 68k can be made to do it. But do we really want to expend the effort to write yet another level two machine, and go through the debugging and.. Or would the software development time be better spent on level three/foour/.../n work, where there is virgin soil to be plowed?

.

Lyle

C2973 CC20 Lyle Johnson (WA7GXD,2973) 10/23/85 1:39 AM L:50
KEYS:/HARUMPH/
A: 10

Gentlemen! Really!

.

(here I digress)

.

There are 14 counties in Northern Ireland, largely Protestant. The rest of the island is primarily Catholic. Both groups are utterly convinced of the rightness of their cause. They have been sniping at each other for a long, long time.

.

And neither side is winning.

.

(or was it digression?)

.

Since I am neither a protocol definer nor a software writer of any particular skill, I am a mugwumpper. (As you may recall in American history in the 19th century, the mugwumps sat on the political fence with their mugs on one side and their wumps on the other.)

.

As a mugwumpper, I am happy to see the progress Phil is making on TCP. 5k! That will fit on any TNC TAPR has produced (read AEA and Heathkit).

.

I am also happy to see Howie's comments on AX25 Level 3 on TNC 2. As I understand things, TCP should run just fine on AX25 L3 (or AX75 or whatever the VC level three is supposed to be called). That is to say, I have been given the impression that levels three and four are supposed to have some sort of well-defined interface so various transport layers can run on various network layers...

.

I suppose there are two other non-technical points I would like to state.

.

The first is that we must be careful to not take ourselves too seriously. Of course, the work that is going on now will have (and is having) dramatic impact on the Amateur landscape. But it IS amateur. The more closely we emulate the commercial approaches (which involved engineering compromises thought best for the COMMERCIAL systems), the more closely we will suffer their same weaknesses. Not too good for disasters, etc., where amateurs get the good will they need to not lose frequencies in Geneva...

.

The second is that amateur radio is thinly disguised anarchy. And that is a healthy state.

.

Packet is technically driven. We don't need emotional IRA-style approaches to "solving" our problems. Four-letter epithets merely indicate a lack of ability to reason and express oneself, a surrender to emotional reaction rather than rational thought.

.

Off my soapbox now.

.
Lyle

C2973 CC21 Phil R. Karn (ka9q,2979) 10/23/85 1:47 AM L:19
KEYS:/ITS MORE LIKE ISRAEL AND LEBANON/68K CONTROLLERS/

Ah hem.

Lyle, as a member of the Digital Committee, have you received a copy of the letter I sent to Paul Rinaldo on protocols? If you haven't, I can summarize the whole thing in one sentence: protocols are important, because they transcend all of the various hardware and software implementations that speak them. That's why its important that we hash out these issues NOW, before inadequately thought-out standards (I was tempted to insert the word "more" in front of "inadequately") are promulgated to thousands of users.

Regarding a 68K NNC: if I have a 68K with a reasonable amount of memory and DMA controllers for the channels, there is NO NEED for outboard processors to do the level 2 functions. They can be accomodated very easily in the 68K itself, and the interfaces are a lot easier to fit when it's all in software. Please don't over-design the hardware; it only makes things harder, not easier, for the software implementers!

Phil

C2974 CC100 Lyle Johnson (WA7GXD,2973) 10/23/85 1:53 AM L:26
KEYS:/NNC UPDATE/BETTER UPDATE TO FOLLOW/THE CABBAGE TRUCK COMES REAL SOOON NOW/

NNC update (sorry it is hasty and brief, will be mmore "formal" ina day or so
.
There was an error in the asrtwork, so the digiatl board and the floppy controller are now one week down (well, 1 and 1/2) and two weeks to go to prototypes. We will get three sets if we are lucky.
.
There was mass confusion on the modem bnoards, so they are just now being cadded. Not in time for southnet2 I'm afraid, on the modems.
.
Just to give you an idea of the magnitude of this project, the CAD costs are right at $9,000 and the 2or three digital prototype boards will cost arount $1,300. Not small potatoes!
.
I will try to post a memory map here in the next couple days so those of you who are interested in proting software to the board can do so.
.
I hope to verify the design and get the artwork modified, wetc, so we can turno out a reasonable number for developers before the end of the year. Ambitious, and may not make it, but that is the target.
.
Still no estimates of costs, but I will post these as soon as I can get a handle onm them. No, we aren't even thinking of recovering the development costs on this one, the 10 grand is support from those of you who bought TNc 2.
.
Lyle

C2987 CC219 Howard Goldstein (N2WX,2987) 10/21/85 11:42 AM L:4
KEYS:/ALMOST FORGOT THIS ONE/K IS FOR KONVERSE/
A: 208

Almost forgot to mention

K is for Konverse -- In Command mode (cmd:), a lone letter "K" is synonymous with CONVerse command.

C2987 CC220 Tom Clark (W3IWI,2976) 10/21/85 11:20 PM L:20
KEYS:/BLACK HOLE MODE/I MAY HAVE TRAPPED THE BUGGER AT LAST/

Howie: I think I caught TNC2 (1.1.1h) in the black hole mode. This time it gave a FRMR, here is the dump (typed manually, since I didn't have a rcv capture file open and was only able to dump it to the printer):

```
FRMR sent:AEDE08
FRMR rcvd:
000: AE6692AE 924062AE 6692AE92 4062AE66 W3IWI 1W3IWI 1W3  .f...@b.f...@b.f
010: 92AE9240 E187AEDE 08               IWI pCWo.          ...@.....
*** CONNECTED TO W3IWI-1 VIA W3IWI
```

In this case, I was in TRANS and had pretty well filled^@the input buffer. Was running MAX 7 and FRACK 1 and everything finally went mute. Even though I was getting no echo, I did a ^C^C^C, set FRACK 5 and then got the FRMR. Again, the only way I can force it to happen, and then only some of the time, is when FRACK is short.

Hope this helps -- Tom

C2987 CC221 Howard Goldstein (N2WX,2987) 10/22/85 6:00 PM L:26
KEYS:/TOM/LYLE/SKIP/BLACK HOLE/CTRL Y/CONV DATA LOSS/

TOM - Ran into something similar to the black hole with the 10.0 mailbox terminal emulator and XFLOW OFF.

With 7 or 8 digipeaters, max 7 and frack 1 the buffer fills rapidly (in transparant with pactime e 1) after a few lines everything freezes up for what feels like forever. It does recover (not like the black hole problem) later when 8 digis end their convulsions and counters start getting updated.

Using the rli B command to send a file while self connected causes a lockup in the both-busy state eventually.

Are you using a terminal program which suppresses characters from the TNC while the flow control is active (like the mailbox)?

LYLE - ctrl-Y tends to work very slowly, especially at high async baud rates. will try to fix it

If possible check the new bank of memory, and/or the checksum for the 1.1.1 version giving you problems ($2f and $7f are both valid early and late checksums).

Using pins 3 and 7 on the rs232 was able to monitor the channel and accept connections okay. Note that the 1.1.1 releases enable hardware rflow always; in default XFLOW ON mode earlier releases used to ignore the state of CTS* on the SIO. The symptoms you mention sound exactly like how some terminal programs act when RS232 DCD isn't true all the time... Haven't seen the inverted CON led problem on the beta board, has anyone else?

SKIP - Still looking, also interested in CPAC/CR/SENDPAC settings during the data loss problem.

73 Howie

C2987 CC222 Lyle Johnson (WA7GXD,2973) 10/23/85 1:46 AM L:20

Howie,

.
I blew it. The problem is the STA LED. Eric connects, has outstanding packets (STA on), then enters cmd: mode and does a pair of rapid Disc. The STA LED stays on. Now, if he connects to another station, the STA remains out of sync. Sorry about the earlier misinformation.

.
Eric is using release 1.1.1 Checksum $7F, 16k bbRAM. If the upper RAM isn't good (lower? the one in the middle socket) it doesn't work at all, so the RAM is good I think. It is a RAM I used as the only 8k RAM in a TNC 2 I was using, so my confidence in it is pretty high. Incidentally, we have about 550 1.1.1 $7F EPROMs programmed now...
.
He is also using a CoCo with 3-wire interface. I will verify that it isn't a four-wire one with some silly flow parameter, but it worked fine with 1.1.0 release version software.
.
Cheers!
.
Lyle

M 6387 J. Gordon Beattie Jr. (N2DSY,2990) 10/23/85 4:01 AM L:11
KEYS:/SOFTNET/56KB MODEM/9600 BAUD MODEM LINK REQUIREMENTS/
TO: (Group 95)

Who made reference to Softnet ? Is there any further information on it ?

Anyone have a 56kb modem planned or working ?

What is a typical 220 MHz circuit configuration for the K9NG 9600 baud modems ? What radios, power levels, preamps, coaxial cables and antennas are being used to cover what kind of path ? What kind of performance is being obtained ? Is it a two point circuit or a multipoint circuit ?
Thanks, Gordon, N2DSY

M 6720 Lyle Johnson (WA7GXD,2973) 10/24/85 12:49 AM L:8
KEYS:/SOFTNET/
A: 6387 TO: (Group 95)

Gordon,

I mentioned it. I will try to get an address for you. They have a multiple 6809-based distributed forth (as in the language) system linked on 402 MHz (non-amateur, and maybe non-existent, but very experimental) with 100 kpbs modems (the modem/radio is the most iffy part).

Lyle

M 6754 Tom Clark (W3IWI,2976) 10/24/85 2:18 AM L:16
KEYS:/PACKET RADIO IN ETHIOPIA/QSP DE WA9FMQ & VITA/
TO: (Group 95)

The following is re-transmitted from AMSAT's TeleMail network:

Posted: Wed Oct 23, 1985 11:56 AM GMT
From: VITA
To: AMSAT, PACSAT
Subj: PACKET IN ETHIOPIA
LATE WORD FROM OUR VITA VOLUNTEER IN ADDIS ABABA IS THAT GOVERNMENT APPROVAL FOR A DEMO PACKET PROJECT/FEASIBILITY STUDY HAS BEEN APPROVED!
AS SOON AS MORE INFO IS KNOWN, I WILL BE CONTACTING THOSE WHO EARLIER THIS YEAR EXPRESSED INTEREST IN GOING OUT THERE FOR A FEW WEEKS TO GET THIS MOVING.
IF ANYBODY ELSE IS INTERESTED IN DOING A VOLUNTEER (BUT EXPENSES PAID) STINT IN ETHIOPIA ON BEHALF OF PACKET AND EMERGENCY COMMUNICATIONS, LET ME KNOW ASAP.
GARY/VITA (WA9FMQ)

C2973 CC22 Phil R. Karn (ka9q,2979) 10/23/85 3:45 AM L:105
KEYS:/TCP ISN'T CLASSIFIED YA KNOW/NETWORK DESIGN/

...ion facilities in existing

Tom: there's no law that says non-US hams can't speak TCP. The ARPA Internet extends to quite a few countries in Europe and to Japan and Korea. We speak TCP/IP regularly across a Telenet/Israeli PTT X.75 connection to Hebrew University Jerusalem (where one of our sabbatical visitors is from), and no member of the International Federation of Protocol Police has yet objected.

It's interesting that you propose addresses based on area codes. You say you want DX contacts, but our telephone numbering plan is limited to North America. I hope you've made provisions for other countries...

Jack: You are mostly correct in your comments regarding layer 2 ("internal") fragmentation of IP datagrams, As I mentioned earlier, AX.25 does not now have the ability to bundle several I-frames together as a single logical "packet", so something sitting above AX.25 level 2 would be needed to perform this function. One possibility is a simple "protocol" that uses reserved characters (with byte stuffing for transparency) to delimit the boundaries of datagrams that extend over several I-frames. Something called "SLIP" (Serial Line Interface Protocol) that does exactly this is already in fairly widespread use. We use it to send IP datagrams over ordinary asynchronous lines, since they also lack packet framing.

If a Level 3 PID were reserved for SLIP, and SLIP is presented with only those incoming I-frames having this PID, then things should work well. Since there's only one IP module atop SLIP performing routing and switching functions, there's no problem with pieces of different datagrams being intermingled. Other (non-IP) traffic having different PIDs can coexist on the same AX.25 level 2 link without interference, since it is the job of level 2 to route each incoming I-frame to the correct level 3 "module" based on its PID.

I'm not sure all this is really necessary, since we can either use IP level ("external") fragmentation, or better yet, allow stations to agree on larger level 2 packet size limits. IP is used to operating on top of many link layers often varying packet size limits. The usual procedure is for TCP to generate datagrams no larger than the maximum packet size of its local interface. If they later have to pass through an interface with a smaller limit, fragmentation occurs at that point and eventually the pieces are reassembled at the final destination; otherwise they arrive in one piece.

I'm curious to see exactly what functions you consider the province of the session level (layer 5). TCP guarantees end-to-end data integrity and provides application multiplexing and demultiplexing, so I don't know what else is needed. Application-specific higher layer protocols (e.g., FTP, the file transfer protocol, and SMTP, the Simple Mail Transfer Protocol) all have "well known" TCP port numbers, so TCP knows how to route incoming connect requests to the appropriate application servers.

Regarding priorities: this is something that IP and TCP do very well. The IP header contains a precedence and class-of-service field so that the network (which deals strictly in IP datagrams) can handle differing priorities on a packet-by-packet basis. The class-of-service field allows the user to control tradeoffs that might be available in the lower-level network used to pass the datagram. There are three bits: low delay, high reliability and high thruput. Setting one or two of these bits indicates that the corresponding service characteristic(s) is/are considered more important than the others. For example, setting the reliability bit might translate into the use of hop-by-hop acknowledgements on a packet radio net, or setting the throughput bit might cause a gateway to route over a high bandwidth (and high delay) satellite channel instead of a narrow but low delay terrestrial link, and so on. Precedence is also available at the TCP level, but this controls access to end-host resources instead of network resources: for example, during an emergency a certain electronic mail system might reject all connect requests below a certain priority level.

You comment about the lack of delivery confirmation facilities in existing

TNCs. Here's where a datagram-oriented transport layer (e.g., TCP) comes in handy. Since TCP itself takes final responsibility for seeing that user data reaches its destination through the use of end-to-end sequencing and acknowledgements, it really has very little use for delivery confirmation from the lower layers. It's *nice*, of course, if the majority of datagrams sent by TCP do in fact reach their destinations, because that minimizes the number of TCP timeouts and retransmissions and hence improves performance, but it seems rather pointless to get a positive indication of delivery from the network for each and every packet when an end-to-end TCP acknowledgment will be forthcoming anyway. It also seems rather pointless for the network to go to heroic measures to absolutely guarantee delivery of every single packet come hell or high water (which it really can't do anyway), since it costs very little for TCP to resend it. The network makes a reasonable effort, and TCP mops up whatever falls through the cracks. As long as the number of datagrams that do get lost is reasonably small, performance is scarcely affected.

An analogy exists in the USPS. Very little mail is sent certified, and this is easy to understand when you consider that the surcharge for certified mail (and the extra complexity of the "interface", i.e., having to go to the PO personally) is greater than the cost of the stamp your correspondent needs to send back an ordinary letter saying he got your mail (and perhaps "piggyback" the acknowledgment to some information of his own, just like TCP.) Certified (or insured) mail is usually used only for items having a high intrinsic value (so "retransmission" is unusually expensive) or mail that is being sent to "uncooperative" receivers (e.g., jury summonses, although around here I get mine from the Sheriff's Deputies!).

The only time TCP can make use of advice from the network is when a datagram CAN'T be delivered for some reason. This is the function performed by ICMP, the Internet Control Message Protocol, and it typically causes TCP to abort a connection attempt (if it receives a "destination unreachable" message) or cut back on the rate it generates traffic (if it gets a "source quench" message indicating network congestion). It makes more sense to me to generate these messages only when something unusual occurs, instead of loading down the network with countless but redundant "all is well" notices.

Phil

C2973 CC23 J. Gordon Beattie Jr. (N2DSY,2990) 10/23/85 4:17 AM L:79
KEYS:/ON AND ON AND...WE HAVE PASSED THE PRODUCTIVE PHASE...SIGH/

In one shot here are my comments on the contributions of the last few weeks.

Jack:

I don't believe that the network will be flooded with a mass of level 3/4/5 connection setup packets.

Level 3/4/5 connections should be opened in one packet unless the upper classes of X.224 are needed in which case they must be done sequentially. This is not bad because the upper classes are used when more than one activity is being conducted between two points. An example of this is bidirectional file transfers and/or conversation.

I am also concerned about the hassle of specific-station PVC tables. This approach is not dissimilar to the hassle we have now with PBBSs needing message routing tables to every other system. An implicit routing scheme based on CCITT X.121, and in the case of the North American region, Area Codes will go a long way to simplify table maintenance.

The "Control" node in the mode you suggested sounds like a further burden. I'm not saying that having some node providing some centralized services is bad, but I don't want to depend on it for actual operation. I am afraid it

either: 1. won't happen, 2. won't be there or 3. will be out of date.
The security and assistance functions are welcome in such a box, but I don't want to get to a stage where we MUST HAVE ONE UP in order to operate.
The download function is VERY DANGEROUS ! I have seen two nodes which had bad code correct one another and then proceed to "correct" switches with healthy code and tables. My suggestion would be to house the software and tables on a system and have the remote switch operator select the code he wants loaded into his machine.

May I suggest that you consider these issues when you think of your PVC based design.

Jack:

Transport protocol support belongs at the endpoints of a circuit. Dumb terminals could benefit from having this protocol support in the TNCs, but dumb terminals usually don't need heavy-duty error checking. Put the effort into a few of the common computers.

Phil and Jack:

The AX.25 Level 2 protocol is not in its point to point form a "datagramme protocol". Yes it has source and destination "addresses" in the header, but they are NOT there for routing purposes. They are there to uniquely identify a link in a common RF domain. In the context of the digipeater environment, the case is different, but again I don't approve (philosophically) of digipeaters either !

As one who has served the amateur community by helping define the protocol, I can say that in its preferable form (NO DIGIPEATERS !):

AX.25 IS NOT A DATAGRAMME PROTOCOL !

Phil:

Regarding your efficiency comparison: read my paper in the proceedings of the last networking conference. My position, and the facts hold up regardless of AMATEUR NETWORK PACKET SIZE. How big do you want Phil ? -- Save your breath: it doesn't matter !

Let us all understand one basic fact: the AX.25 protocol suite as we designed it (over three years ago) and as it is currently being implemented consists of a link protocol (AX.25 Level 2, version 2) AND a network protocol (CCITT X.25 Level 3, 1984).

Bravo to Howie in C2973 CC11 !

Brevity is not one of my stronger characteristics, but it seems that I've come farther than some...G'day !

C2973 CC24 J. Gordon Beattie Jr. (N2DSY,2990) 10/23/85 4:25 AM L:8
KEYS:/THE DOD IS LOOKING FARTHER AHEAD THAN SOME FOLKS CAN HANDLE/KEEPER OF THE DEAD SCROLLS/

4Phil,
I guess you need weekly reminders: The DoD is going to replace TCP with CCITT X.224. Let's all remember this and take turns reminding Phil.
There has been much discussion over painful conversions to "superior"

protocols such as the DoD suite. Isn't it odd that the conversion from TCP to CCITT X.224 is NEVER mentioned in such discussions ?

C2973 CC25 Thomas A. Moulton (W2VY,995) 10/23/85 9:42 AM L:10
KEYS:/LEVEL 3 IS VERY MUCH NEEDED/
A: 18

Jack,
We have a best a shakey point to point protocol, DIGIPEATERS are a KLUDGE and should not be used much in the future, by running X.25 Level 3 we will get relaible point to point links, spanning as many stations as is needed, not being limited to 2 + 8 digipeaters, the throughput will be higher bacause there will be hop by hop acks (between each switch) so no more long delays due to your frame being trashed on the first hop of 8 (or even 2)
If you get carried away you can call almost any protocol a datagramme, incorrectly
Yes there are two addresses, they specify the direction, in our case we must specify the call signs, A->B and B->A

C2973 CC26 Thomas A. Moulton (W2VY,995) 10/23/85 10:09 AM L:9
A: 22

What I meant by using area codes was to indirectly reference a paper from the last conference, 4 digits for country code and then the remaining digits are for national definition, in North America; area code (3 digits) and number 7 will work just fine, other countries use city codes and a local code before the number which will work also, after all we can call them on the phone too.

I do remember there were some countires that would only allow HAMS, (aka Amateur Radio Operators via RF) using internationally recognized protocols.

C2973 CC27 Howard Goldstein (N2WX,2987) 10/23/85 1:41 PM L:17
KEYS:/MULTIPLEXING/X.25/X.75/COMMERCIAL VC/
A: 15

JACK - X.25 network layer is especially nice at the user's TNC/PAD since it allows consistent way to multiplex different virtual circuits across a single link regardless of their destination. X.25 13 in particular is nice for this since it should be easy to implement on existing TNCs. Between two DTEs it also provides multiplexing (at least the way I have it.. what you're called as far as DTE/DCE is determined at call setup time)

PHIL -- As you saw in the TNC 2 beta test conference I don't have X.75 specs, so I'm interested in the differences between an X.25 interface implemenation and an X.75 network implementation. For what its worth - provided there's a routing mechanism - X.25 running between two DTE's seperated by n DCE's works quite well.

No one has yet discussed why the commercial networks use VCs. Does it have something to do with the cost of providing useful and reliable service?

73 Howie

C2973 CC28 Phil R. Karn (ka9q,2979) 10/23/85 3:36 PM L:53
(ORIG.) 10/23/85 3:19 PM L:53
KEYS:/I HAVE PLENTY OF PATIENCE/

Gordon, I don't want to get too bogged down in semantics, but I suggest you read chapter 5 of Tanenbaum's book (Computer Networks). A datagram protocol is one that has both the source and destination addresses on

each packet; that's it. The link header of Ethernet is about as similar as anything you'll find to the lower part of AX.25, and everyone certainly considers Ethernet to be a datagram network. You are just bothered by the irony of having participated in the design of a datagram protocol, but a rose by any other name...

My point about header efficiency is to show that there are two variables in the equation (header size AND data size) and that the effect of a larger header can be made negligibly small (but not zero) by increasing the data packet size.

I thought I posted the DoD position paper on TP-4 vs TCP a few months ago. This was in response to the NRC report. Their position was that it is currently infeasable to specify TP-4 as a co-standard with TCP, or even to set a specific date at which this could happen. The reason? The standards are full of unresolved holes and missing functions, and commercial implementations are nearly non-existent. When (or if) they are plugged and TP-4 becomes available in something other than paper, DoD will reconsider using it and so should we. In the meantime, I think TCP has quite a few years of life left in it.

Gee, it seems to me that if we specify TCP as the standard transport protocol for amateur packet radio, then it will become "internationally recognized" in the same way that AX.25 did (the name is really a misnomer, because it isn't a compatible form of X.25 in the sense that BX.25 is). Or are there countries that don't permit the use of AX.25 TNCs because CCITT didn't publish the exact spec? I think a much bigger problem with other countries is the use of automatic repeating and linking by ordinary hams; certainly they could care less about the specific protocol.

Howie: commercial public data networks PROVIDE a virtual circuit SERVICE; they do not necessary use them internally. Telenet and Tymnet do, but other networks (e.g., Canada's Datapac) use their own datagram protocols inside the network. Why do the PDNs provide only a VC service? A couple of reasons. First, the protocol was decreed by the CCITT, an "old-guard" organization of telephony people who have had the "circuit switching" notion ingrained into them from day one. Second, 99+% of X.25 users do nothing with it but connect their dumb terminals through PADs and access their big IBM mainframes. In such a limited application ("remote access" as compared with the more general "resource sharing", which is characterized by more computer-to-computer kinds of applications) the X.25 straitjacket isn't much of a hindrance and the lower per-packet overhead is an advantage because most people program their PADs to send single-character packets. If you try to use X.25 for host-to-host (i.e., computer-to-computer) applications, as we do, you start running into X.25's limitations very quickly. You are forced to use more sophisticated higher layer protocols on top of X.25, making many of the "features" of the latter so much excess baggage.

Phil

C2987 CC223 Howard Goldstein (N2WX,2987) 10/23/85 8:46 PM L:12
KEYS:/FLOW/INDUCED FAILURES/
A: 222

Lyle - The earlier lettered releases assumed software only flow control until told otherwise. Wasn't TNC 1 compatable...$7F does use hardware flow all the time. It shouldnt affect echoed stuff though and any mode that disables the TNC echoing is bogus

I tried screwing up the 2 ram chips a few different ways, the least damaging of which (no RAM at 0xC000-0xDFFF) left the TNC flashing its lights and keying the PTT smartly - to the beat of distant drums, no doubt - and it echoed, but otherwise showed no signs of life in terms of prompts or anything else. The other induced "failures" left the board completely dumb (1.1.1 $7F 16K)

Please send me a message if you would like to help out.
(if this doesn't work I may have to ask for a justification statement from each of you... I really don't want to wade through all the mail tho...)

M 6319 Jack Brindle (WA4FIB,2963) 10/23/85 12:21 AM L:16
KEYS:/OCTOBER DOES NOT MEAN NOVEMBER./
TO: (Group 95)

To: all MacPacket users awaiting TNC2term.
From: Jack Brindle, WA4FIB

We have decided to delay shipping MacPacket/TNC2term until at least December 1. The reason is the lack of hardware to test with. TAPR promises a TNC2 REAL SOON NOW... I am told that will be after November 1. I am trying to introduce the new version in unison with a major MacPacket upgrade, now scheduled for early December.

I understand TAPR's desire to ship the latest and greatest, but they should at least keep the folks they are holding orders for posted on their progress. Many TNC2 orderers are under the impression that TNC2s were to go out on the 15th. Anyway, I KNOW it is a volunteer organization and we must wait for the volunteers (we are all familiar with that, since packet radio has been developed by volunteers). In the meantime, those awaiting TNC2term, please have patience. Things can only move so fast.
73, Jack B.

M 6350 Lyle Johnson (WA7GXD,2973) 10/23/85 1:22 AM L:27
KEYS:/TNC 2 STATUS REPORT/
TO: WA7GXD, (Group 95)

TNC 2 update.
.
For those who have not called TAPR recently, the following is the TNC 2 situation.
.
TNC 2 through 511 shipped. We were shorted over 100 pc boards for our september delivery and, after careful consideration, changed PC houses. This has cost us about $2,000 in new setup charges and about $8.00 per board in purchasing costs.
.
The TNC 2 has been relayed out, the new ones have double memory and even better software.
.
Cabinet front and rear panels were sent out to TNC 2 Rev 1 owners.
.
By Thursday ofd this week (24 october) we will have everything for nearly 500 TNC 2s except cabinet front panels, due sometime next week.
.
We will ship TNC 2 Rev 2 without front panels, and send the front panels when they arrive. We should have up to order 900-950 shiupped out early the week of the 28th (we want to make it October if we can!!!).
.
Most units will be all CMOS. Extra RAM, extra EPROM< CMOS, new boards, now 5 LEDs (PTT added) all at no increade in cost. Thye manuals have been completely edited and reprinted as well.
.
Thank you for your patience. The wheels really are turning...

M 6353 Lyle Johnson (WA7GXD,2973) 10/23/85 1:28 AM L:1
KEYS:/PSR/
TO: (Group 95), WA7GXD

October PSR Quarterly is at the printers. Should go out in October!!!

Feel rather strange with 550 blown chips and 2-3 ?bugs...

73, Howie "There's another bag of coffee in the cabinet over the sink" G.

C2973 CC29 Phil R. Karn (ka9q,2979) 10/24/85 3:14 PM L:19
KEYS:/WHAT THIS DISCUSSION NEEDS IS A GOOD DOSE OF COMIC RELIEF/

I forwarded an archive of the Protowars for the past week or two to Brian Kantor, WB6YCT out at UCSD. I thought you might be amused by his reply:

From brian@SDCSVAX.ARPA Thu Oct 24 14:41:16 1985
Date: Thu, 24 Oct 85 11:20:44 PDT
From: brian@SDCSVAX.ARPA (Brian Kantor)
To: karn@mouton.arpa
Subject: Re: EIES packet radio protocol discussions

You are hereby morally encouraged. Also immorally. Remember, we're right and they're wrong, and they're also too uninformed to know that they're wrong. Goodness and Right shall triumph over Evil.

Feel better? When I've got .3 second to myself I'll try to marshall some arguments, but you seem to be carrying the ball quite well.

Illegitimus non Carborundum est.
- brian

C2973 CC30 Jack Brindle (WA4FIB,2963) 10/25/85 2:20 AM L:166

To All;

After watching DR Net for several weeks, I jumped in and restarted the protocol arguments specifically to bring myself up on the current thinking in the various "camps" around. I have noted several things. First, there are two groups, one centerred around datagrams chiefly represented here by Phil. The other supports Virtual Call type virtual circuits (while disavowing PVCs), and contains Tom, Howie and Gordon. Both sides have legitimate arguments, and they certainly like to argue. I wonder whether the two groups fully understand the stands the other group takes, and more importantly why. It is interesting, though. I just want to make sure that we don't go back to arguing instead of continuing on our projects.

Having said that, I will now dive back into the discussion. First off on the 68000 board issue. I am not averse to using an outboard processor to perform individual layer functions. This might be necessary if we want to attain high throughput rates (say T1). However, there is a major problem with current TNC implimentations. The WA8DED code is the only code available at present that has a decent computer interface, and runs in an "off the shelf" TNC. We continue to develop TNCs with dumb terminal type interfaces, yet the vast majority of packeteers have their TNCs attached to computers. It would be far better to allow the computer to perform the higher level functions and have the TNC perfrom the layer 2 functions it is intended for. If you wish to investigate this, try interfacing a Kantronics or TNC2 to a computer. You will quickly find out why I am making major changes in my MacPacket products (no announcement yet on that).

To be specific on what I would like to see, first I would like an interface that allows me to poll for data and status. Take a look at the way floppy disk controllers work. We should follow this example (This is one of the reasons that the Mac is so nice to program, its drivers are set up logically in this format!). Instead of throwing out data uncontrollably, the TNC should only speak when it is asked to. This also allows for a better seperation of commands and data. I should also be allowed a free dialog of the TNC's status - like when it sends a packet (or even better, when it receives the ack for that packet). This allows me to better perform the upper layer functions. For example, it is nearly impossible to add priority to a current TNC. (Yes, I know this is a layer 3 function). Since current TNCs buffer data from the host, they may have many packets stacked

on their outbound queue. If the system wishes to send a higher priority packet, it can only tack it on to the end of the list. One way out of this would be to only send one packet at a time to the TNC. But I would need layer 2 delivery confirmation to perform this function. Thus we are back at square 1. Well, how important is priority. Suppose two stations are in a QSO. They have my session layer code that allows multiple sessions. One operator requests a file transfer. The other agrees, a new session is set up and the file transfer proceeds. With no prioritization, the data file fills up the outbound data pipe. The two operators continue their QSO, but find the going rough as their exchanges must wait in line behind the file transfer. A better situation occurs if the QSO is given higher priority than the file xfer so that packets from the keyboard are enqueued at the front of the line so that the QSO proceeds almost as if the file xfer were not happening.

Another reason for needing delivery confirmation at each layer is data integrity itself. Each layer must hold the outbound data (or at least a pointer to it) until it receives an ack from the peer layer. For example, suppose I send a packet. It comes in at layer 7, and proceeds downwards. Layer 4 adds its info, then passes it on to layer 3, etc. If L4 were to delete its copy of the data after it was sent to L3, then what would happen if the delivery were not made? Thus it needs to keep at least a pointer to the packet text, and the packet must remain intact, until a peer confirmation is received. This peer confirmation, of course, comes from the layer beneath the current layer. The lower layer also must report delivery to show the upper layer that it is functioning properly. Note that in our point to point system, this acknowledgement is equivalent to an end-to-end ack, and would release the packet as far as L2 is concerned, and would allow the session layer to also release the packet (Confirmed packets would also require a peer layer ack). The confirmation in a network system would simply show delivery to the next node in the network. It is important to realize, though, that these confirmations occur between each successive layer.

This also points out why it would probably be better to build the protocol for each layer into the node controller. The node must operate in a homogeneous fashion or we will have problems from a loss in data integrity. If we can get a system using NNCs as front ends for a larger processor, that is fine. In fact, that is the approach used by the larger Noed controllers (reference Paradyne's Pixnet XL line with processor boards using Z8000s for each layer). My 68000 NNC design does not preclude this approach. In fact, its memory design could even handle twin 68000 coprocessors interactively sharing the bus (what a programming nightmare).

Next subject, long packet fragmentation. Why not just shorten the packet length to a suitable size. We have a unique transfer media. Ethernet systems using Coax cable do not have the noise, QSB, and QRM problems we have with radio. Thus they can get away with long data packets. We have BER problems as it is. Try a test sometimes over a marginal, but useable, link. Shorter packets get through far better than long ones. The reason, of course, is that short packets have less of a chance of getting blasted than long ones. This is one reason that the 8 call digi field is idiotic. That excess baggage causes extra problems in getting good packets through. The answer is not to lengthen packets, but to either condense them (this is a good use for layer 6 - to compress data strings by, for example, changing multiple occurances of characters (a row of blanks) into a count & the char). We can actually increase our throughput by shortening the packets! AX.25 has set a max of 256 octets. This seems reasonable to me. In fact, the MacPacket file transfer protocol blocks data at 128 bytes. Add to this overhead from the various layers, and we are up towards 200 bytes per packet. Of course, the packet shrinks as we remove digis! This information would have a far better chance of getting through a marginal link than a packet with 1024 byte data blocks. Nuff said.

We all agree that digipeaters are an abheration (sp?) on the packet scene. Well, are they? They certainly have made our current operations useful. The big problem is that they use the stupid digipeater fields instead of real network routing. Well, folks, that is why we are here arguing. I suspect, though, that even after we get our networks up we will still have a use for the digis. They are very useful to link local stations that are

just out of range of each other. For example, I can connect to packeteers all over the Florida west central coast using an IC2AT running 150 milliwatts, and a 1/4 wave whip antenna. Why? Because KC2FF-7 is only 1/4 mile away! Without it I would not be able to operate on packet! It wouldn't make sense to use the resource of the big local node just to talk locally. Even better, we may need the things to allow stations that are outside the rangge of a network node to access the network. I suspect this will be their real use. It will be interesting to watch as these things develop.

One thing that we should all remember is that we must make things easy for present and new users to play with the system. We have a lot of new packeteers who would not be on if not for the Kantronics TNC. These folks have decided to buy a new gadget and find out what the noise is all about. If we build a network that is difficult to use, or requires major modifications to their station equipment (The very simple Kantronics units may not be able to handle much more), then a separate, parallel system will sprout. This, of course, would undermine all our efforts. We must leave a place in our systems for these people. Their TNCs may eventually go away, but it will take quite a while. The vendors are not designing state of the art stuff, but they are making money. One example, the new Kantronics Version 1.6 ROM does not speak AX25L2V2. Instead it just rehashes the old code. They did add a "check" function which, in the absense of activity on the connection, will send SABMs to the other station. Prepare to see lots of "Connected to..." or "Link Reset..." messages caused by people using these units. But, to allow the use of these things, we must design the network interface for them. If I have the full network code in my machine I should be able to use it. Otherwise, I should be presented with information to allow me to use the system.

I foresee an interface that would present the Kantronics user with a menu for the routing function, and perform layer 4 functions for him. You might sya, "What a kludge", and you are right, but it would work.

Lastly, a comment on TCP/IP/Ethernet. The Ethernet protocols do not follow the OSI model. I understand that there has been an attempt to bring TCP/IP under the OSI model, but I do see that it did not completely happen. For example, XNS "sockets" (which I presume also exist in TCP) are actually OSI layer 5 functions. Thus, to answer your question Phil, You may not need my session layer. But you had better have its functions! The TCP/IP functions overlap the OSI model in many areas, creating confusion for thos not understanding both systems. In Xerox XNS/Ethernet, layer 0 handles OSI layers 1,2, and part of 3. This layer is known as the Ethernet layer. XNS then handles internet routing, the transport and higher OSI protocols. Of course, it still maintains an application type layer where the user system interfaces. I assume that TCP/IP is not much different due to their similar roots.

For those wondering about my session protocol, it allows multiple sessions within a computer. This would allow a multi-user computer to carry on a seperate session for each user on the system talking over the network, or allow seperation for many stations logged into a multi-user bbs. It also allows for multiple file transfers and qsos on MacPacket system,s. It works to seperate different dtat streams, so that whether the lower layers can handle multiple connects or not, the session layer allows that. When connected to one station, the user can both QSO and send file(s) simultaneously. When connected to sebveral stations, the user can talk to each, and transfer files simultaneously. It may be slow at 1200 baud, but it works. The streams are seperated into sessions, which are identified by a unique session id. The session layer relies on the lower layers for packet data integrity, but adds checkpointing in the case of file transfers. This is to ensure that file blocks are properly sequenced (this would seem to be a layer 4 domain, but layer 5 is uniquely set to perform this function for each individual data stream. Layer 4 sequences for all inbound packets, while layer 5 performs the function for its particular stream).

I will shortly release more info about the protocol, and plan to give a talk on it (hopefully with a fully fuctional MacPacket demo) at the Southnet Packetfest in Atlanta next month (Y'all come now, hear?).

This is enough for tonight. More later. 73, Jack B.

C2987 CC224 Lyle Johnson (WA7GXD,2973) 10/24/85 1:54 PM L:34

KEYS:/RS232 FLOW/STA LED CURED/EGGS DELIVERED HERE(FPOR MY FACE)/

Howie,

Oops! The STA LED problem was a release 1.1.0 problem. Eric verifies that it isn't a problem with release 1.1.1. This was my misunderstanding, my apologies.

Here is the straight dope on the flow control problem (Eric is sitting at my right elbow as I key this in).

During power up, while the CON led is illuminated, the TNC 2 sign-on message comes out normally (baud rate to computer is 1200 baud). When the CON led extinguishes, output originated by the TNC ceases. Characters typed in echo normally. Since he is in cmd: mode at power up, any command issued is executed normally, but the response indicating the command was acted upon (eg, value was xxxx) does not appear, nor does the command prompt (cmd:).

If he then connects to a packet sataion (including a self-connect), and the CON led comes on, then all the command mode information that didn't appear as per above, suddenly dumps to the screen and the unit oiperates normally.

Upon disconnect, two asterisks appear, then nothing (except echoed characters) appears until the next CON led illumination.

Eric is using a Radio Shack Color Computer (says color but the case is mostly black and white) with a 3-wire interface -- txd, rxd and ground. There is no othe rpin connected at either end of the rs232 cable. This is the same cable connection that he has used sith TNC 2 during beta test, with other TNC 2
software, with TNC 1 and with the Beta board. All of which work without displaying
this characteristic.

End of report. Keep the caffiene handy. Your faityhful, though sometimes not-totally-accurate scribe,
LJ

C2987 CC225 Howard Goldstein (N2WX,2987) 10/24/85 3:24 PM L:12
KEYS:/DATA LOSS EXISTS/FIX COMING/

to: beta
fm: howie
re: data loss problem found
dist: closed

First Skip thanks for sending the DISP dump along. I have indeed been able to duplicate the CONV mode data loss when ECHO is OFF.

This is a serious bug! I think I have a fix that seems to work and will upload it later today or tonight after class. 73 Howie

C2987 CC226 Howard Goldstein (N2WX,2987) 10/24/85 3:40 PM L:8
KEYS:/SEND EGGS HERE LYLE/THEY BELONG ON MY FACE NOT YOURS/FANTOM FLOW?/
A: 224

Lyle - If Eric can, ask him to check the voltages at U9b pin 3 and U21 pin 23 when the board "isn't on speaking terms". I suspect there's a short that's keeping the U21-p23 pin high when the CON LED is off, and it's flowing off the xmission from the TNC to the CoCo

PS this problem would not have been noticable on earlier versions that didn't recognize hardware flow on default power up

73 Howie

C2987 CC227 Tom Clark (W3IWI,2976) 10/25/85 12:51 AM L:7
KEYS:/OMISSION FROM MANUAL/AWLEN & PARITY SET ONLY ON POWER-UP/

Had a user (our friend WB6RQN) report he was unable to change PARITY and AWLEN. I recalled that these parameters were set only on power-up and was going to reference the manual to him. Scoured the manual high and low and couldn't find ANY reference to it! Looks like this should be flagged in subsequent errata! Howie -- is there any real reason for not allowing AWLEN and PARITY as immediate commands? Might be easier to fix the code than to fix the documentation!

C2987 CC228 Lyle Johnson (WA7GXD,2973) 10/25/85 1:06 AM L:5
KEYS:/MISCELLANEOUS INFO/SIGH/

unfortunately, the eproms are burned, kits are packaged and errata is printed. I can do another errata, but I think the 1.1.1 code we have now is the release stuff we will have to use. kits start shipping again tomorrow or monday at the latest 9sigh). I will have eric check voltages. (errata runs 13 pages this time!). Lyle

M 7139 Harold Price (NK6K,2972) 10/25/85 5:18 AM L:151
KEYS:/HOW MANY TNCS ARE THERE?/
TO: (Group 95)

From: Harold Price, NK6K
Dept. of Prognostication
To: ALL
Re: Fun with numbers
Dist: Open

How many TNCs are there? (And how many will there be?)

Aside from being an answer to a trivia question, it is one of many items that must be considered by network planners.

The installed base, as of the end of October, 1985, by my estimate is: 8750.

This number is based on actual TAPR numbers (3700), and information from other vendors. The other vendor numbers are a blend of (1) their numbers (high), (2) their competitor's estimate (low), and (3) various other information sources. Results are checked for credibility against published packet census data from various locations. The number includes TNCs shipped overseas, but does not include TNCs manufactured offshore, nor does it include bare board TNC-1s manufactured and sold worldwide under TAPR's OEM agreements.

These numbers are within 15 percent, and are conservative. It isn't sporting to break out the list by vendor.

Other interesting facts about these numbers:

An estimate of the money spent on TNCs so far is $1,892,000.00 (One million, eight hundred ninety-two thousand dollars).

The Heath and Kantronics units have only been on available since April 1985, the TAPR TNC-2 since August 1985. A little more than half of all TNCs in existence have been sold in just the last 6 months.

How long have TNCs been around?

Some major amateur digital milestones.

Sept 1978	non-baudot digital transmissions made legal in Canada. Digital experimentation begins.
Jan 1979	VADCG group formed. This group produced the VADCG TNC, some are still in use today.
Summer 1979	Work beings in Ottawa and Montreal. Total North American digital usersj Ness than 30.
March 1980	Ascii data legalized in U.S. Canadian missionaries armed with VADCG TNCs and software cross the boarder.
Dec 1980	First U.S. digipeater goes on the air in San Francisco, it uses homebrew hardware and software based on the VADCG protocol (now called V1).
1981	First great packet diaspora begins. VADCG distributes PC boards. Homebrew systems are developed. Most areas standardize on 1200 baud bell 202 modems and VADCG compatible hardware. Locally maintained software versions in San Francisco, Washington DC, Vancouver, and elsewhere begin to diverge.
Oct 1982.	AMSAT and AMRAD host another in a series of meetings to solve the divergence problem by developing a protocol standard. Other major goals include the desire to support more than the 32, 64, or 128 users allowed by then current V1 implementations. The AX.25 standard is born. Total North American digital users: no more than 200.
Jan 1983.	After several months of design and testing, TAPR produces 170 assembled and tested TNCs.
Oct 1983.	TAPR TNC kit (now called TNC-1) is beta tested by 19 users.
Dec 1983.	200 TNC-1 kits are shipped. In the mean time, more VADCG boards were assembled. GLB takes out first ad in QST for an assembled and tested unit. Total TNCs: about 650.
1984.	TAPR begins to ship TNC-1 in bulk. They ship an average of 120 TNCs/month for the next 15 months. AEA announces an assembled TAPR TNC-1 clone at the Dayton Hamvention. Packet hits the big time when Lyle, WA7GXD wins the Dayton Technical Excellence Award, he accepts on behalf of packet radio and TAPR. AEA legitimizes packet by placing the first full page ads in the big ham magazines. At the end of 1984 there are more than 2500 TNCs.
1985.	Heath announces HD4040 TNC-1 clone kit. Begins shipping in April, sells out first 500 in three weeks. Kantronics announces "Packet Communicator". TAPR announces TNC-2. GLB announces PK1L. AEA announces PK-64 and PK-80. For a time in August, most of the packet industry is "sold out". with

demand far exceeding production.

Summary:

1982 200 tncs.
1983 650 tncs.
1984 2500 tncs.
1985 10000 tncs. (projected, 8750 tncs through Oct 1985.)
1986 ????

Where will we be in 1986?

AEA has announced the PK-64. Targeted are all those C-64s setting in closets out in Ham-Land, it is the first low priced unit to include Packet, RTTY, and AMTOR in the same box. Industry-wide, expect at least another 5000 TNCs by the next ARRL networking conference in March, 1986. That's about 14,000 TNCs.

How far can we go?

The 1985 Callbook lists about 460,000 amateurs in North America. A study commissioned by the ARRL a few years ago found that half of the amateurs polled considered themselves "active". Perhaps a better indication for our purposes is the number of RTTY units sold during the big computer-RTTY boom of a few years ago. A discussion with two vendors at the Dallas hamvention early this year and with another vendor this week resulted in a guess of 50,000 units sold to 30,000 individuals.

I believe our growth will peak somewhere between this figure and the total number of 2 meter HTs. I can't get a good guess for that last figure, anyone want to take a stab?

I also believe that for Amateur Radio to survive in the long run, we'll need new blood. A large percentage of new hams will be interested in digital radio, many will be drawn to our hobby solely for its digital aspects. So in the end, who can say how far we'll go?

Next issue: What has this got to do with network planning? or
Network gateways: love 'em or hate 'em, we'll have 'em.

M 7422 JEFF WARD (ARRL,2977) 10/25/85 10:21 PM L:11
KEYS:/SOFTNET/
TO: (Group 95)

RE: Softnet and their fast modem

The only sources of information that I have ever had on the Swedish SOFTNET group were therir newsletters. Their address was in a couple of issues of Gateway, and I am sure that if you send them a letter they will repl.y. They have kits available for complete SOFTNET nodes, and I think that they are also offering the nodes (which include radios) assembled and tested for a reasonable price. I am surprised that there has been no experimentation with their stuff in the U.S. Any FORTH maniacs out there?

C2973 CC31 Harold Price (NK6K,2972) 10/25/85 8:19 PM L:170
KEYS:/MAGIC BOXES/

From: Harold, NK6K
To: layer three group.
Re: network access via magic box
Dist: open

Its time for my two cents worth, even at risk of being called the

captain of the don't care crowd by Phil.

Unfortunately for the theoretical bunch, amateur radio is the real world. It is in some ways a bit more real than the bastions of TCP/IP or X.25, since we don't get large infusions of public money or that dirty "profit" money. We have to pay for our resources dearly, and employ large amounts of slave labor.

Packet Radio in interesting in that it has two groups of participants, those who are interested in the data that is sent, and those who are interested in the way it is sent. I've discussed this before. Each group supports the other. This gives those of us in the network planning area a problem, one of supporting "users" while playing with the network.

While some amount of arguing and theoretical gnashing is good, there are some realities being ignored here.

Here is one such reality:

```
Fig 1.
Unmodified -(AX.25)--Magic Link---(backbone net----Magic Link -(AX.25)--Unmod
AX.25 TNC            Box          protocol)        Box                  TNC
```

which some of you will recognize from one of Phil's messages.

There are now, and will always be, more "unmodified AX.25 TNCs" than anything else in the network. The label should actually be "unmodifiable AX.25 TNC", because there are too many of them to update easily. In another message I pointed out that there are 8750 TNCs today, with at least 14,000 due by the 5th ARRL networking conference in March, 1986. 14,000 TNCs aren't going to have a network layer and a transport layer added to them any time soon. If we want the majority of users to be able to connect to the network, they're going to have to access it through a network entry node, or "magic link box".

Some people have said that figure 1 is undesirable, and to some extent I agree. In a perfect world it would be nice to have each individual directly connected to the network. Some people have gone further and said that figure 1 is unrealistic, unreliable, unimplementable, or outright criminal. I disagree. If we change some of the labels on Fig 1, we get this:

```
Fig 2.
dumb terminal--------Magic Link---(backbone net----Magic Link-----------HOST
                     Box          protocol)        Box
```

a further mapping results in:

```
Fig 3.
your terminal------telenet node---(telenet)-------telenet node----------EIES
```

Not only is Fig 3 (and therefore fig 1) implementable, it got you this message. Yes, this is an old way to do things, yes, there is no end to end guarantee that data sent from EIES will appear on your terminal. But it does the job. And given the realities of packet today (8750 unmodified TNCs), we'll end up implementing Fig 1 anyway. Is that bad? Read on.

The biggest problem I see with the theoretical blasting going on now is a lack of a defined goal. What kind of a network is envisioned? It appears that

```
Fig 4.
end user tnc ------------(network)--------- end user tnc
```

```
                                                    or
                                                    host
```

is the implied goal.

Laudable, but its not within our grasp if we intend to support the user base any time soon.

Fig 5.

```
TNC1                                             TNC7001
.                                                .
.---------------------(NETWORK)-------------------
.                                                .
TNC7000                                          TNC14000
          |                        |
      interface A              interface A
```

In fig 5, 14000 TNCs (march, 1986) must be updated, and kept up to date, with protocol and interface a. This is unlikely in the extreme. To be fair, it is also taking the fig 4 model to the extreme, but it seems that this position has been expressed lately.

Fig 6.

```
TNC1                                                                TNC7001
.                                                                   .
.----------(NETWORK node)-----(network)-----(network node)--------
.                                                                   .
TNC7000                                                             TNC14000
       |                 |              |                  |
   interface B      interface A     interface A        interface B
```

At last year's meeting of the digital committee, the group decided that fig 6 was more likely because interface B could be fixed at what was available (AX.25 on 14,000 units), and interface A could be diddled around at will because it exists on far fewer units.

Someone recently lamented "Woe is us, how will we tell the magic link box who we want to talk to?". On the network you used to read this message, interface b is an ascii serial stream, and you used x.3, x.28, and x.29 to get the network to do your bidding. On our network, interface B is an AX.25 connection, X.3, X.28, and X.29 can be used to handle PAD control.

To make the point again, it has been argued that fig 6 (a mapped fig 1) is unworkable. It got you this message, and it is far better than a chain of digipeaters. And it's a good thing that it is implementable, because with an installed base of 14,000 by March we haven't got many other choices than to offer network service via a gateway.

I'm not advocating a freeze on network access development to a level two interface and a network diagram as in figure 1. But for several reasons it is the easier path to take first:

o User pressure. Users, including us, need a more reliable bulk file transfer system than we have now. The simulated earthquake and the real fires in southern California have shown the limitations of our current system. There is real need for something now.

o User inertia. There are 8750 TNCs now, 14,000 by March. The majority won't be upgraded to layer3/4 boxes soon, maybe never.

o Staged growth. Fig 5 is an all or nothing proposition. Fig 6 allows for distributed development and partitioned growth. The network will grow bit by bit, with need forcing innovation. Sure, more energy is expended in the long run, but the increments are manageable. Amateur Radio can't afford to "buy" our network outright, we can only afford the installment plan. It therefore makes sense that we design to a model that will work by installments.

Note that this discussion has not touched on VC vs datagrams. The final choice will be invisible to most users anyway. As time goes by, hosts will connect directly to the network rather than through a gateway node. Users will too, but not soon, and not most of them.

Summary.

The purpose of this document has been to counter some of the comments that figure 1 is undesirable or unimplementable. It discusses figure 1 as inevitable based on current user numbers and devices, and desirable because of the possibility of incremental implementation. It uses TELENET and EIES as an example of a figure 1 implementation that provides a data transfer service far better than what we can now offer with a string of digipeaters, even though it does not maximize "elegance".

C2973 CC32 Lyle Johnson (WA7GXD,2973) 10/25/85 11:53 PM L:38
KEYS:/NETWORKS/USERS/NUMBERS/

Of user interfaces and the "majority" of users.

It is true that most TNCs are connected to computers. This is NOT because most people now coming into packet radio are dedicated computer types. It is because a C64 is cheaper than a VT100.

A LARGE portion of TAPR's mail consistsa of: What is RS232? How can I connmect my VIC 20 to a TNC? What program do I use to talk to the TNC? What is ASCII? What is BASIC? Do I need to know how to program to use packet radio? etc.

Go to any ham radio conevention where a couple thousand people attend. Hang around the packet booth. Ignore the 8 people that spend the whole convention at the booth (no, not ignore them, but ignore them for now...) and you will see that the vast majority of the gawkers and enquirers are beign handled by the other 3 people at the booth who are not part of the 8 who engage in hard-core technical discussions. And the 3 people are answering the aboce questions for joe and mary HT.

And Joe and Mary, the ones that hang out on the locasl 2-meter machjine and talk about being "nearly destinated" are the same Joe and Mary that are going to be the largest percentage of user types in any packet network in the next year.

The hi-tech typess are already saturated in packet. Most of you have more than one TNC and many of you have been involved in packet since the early days of 1982 and before. You understand the issues, or probably do.

Joe and Mary want to communicate. "I am nealry destianted, 10-4." will be the majority of the error-free traffic ahndled.

Think about it. 10,000 today (well, 10,000 by the middle of Nocember), 14,000 by ARRL number 5. 25,000 by Christmas 1986. maybe more. probably not less. Most with C64s.

Fig 6. here we come!

Lyle

C2973 CC33 Harold Price (NK6K,2972) 10/26/85 1:10 AM L:127
KEYS:/NETWORK PLANNING/

To: Level 3 group
Fm: Harold, NK6K
Re: Just what are we doing?
Dist: open

It strikes me, whilst perusing the recent goings on, that I don't have a clear idea of what sort of problem is being solved here. When most organizations sit down to begin network planning, they have a goal in mind. "We're going to connect the 4 PCs down the hall with the printer here and the bar code reader over there", or "we're going to connect the central office with the 14 district sales offices", or "We're going to wire the new building so that each of the 100 currently planned user stations can transfer files and mail between each other".

The amateur thinking in some parts seems to be

> We need to connect computers
> Therefore, we need a network
> (blank) is a pretty good network so
> let's implement blank.

Thus, we get offers to get sent copies of the Complete ARPANET Standards, or the X.? fascicles, or whatever. We get comments like "We're doing ethernet without the cable". I'm guilty too.

That leaves us where we've been for 3 years. Trying to get an entire network standard to be accepted and implemented in one fell swoop.

It's hard to not want it all. Its hard to talk about the amateur network to your non-ham professional peers. They say

> "(sniff), protocol (blank) has solved all of the problems you are ever likely to encounter. We here at (blank) believe that the (blank) protocol is best for all needs, and certainly best for your piddling network. If you aren't pushing for (blank) to be implemented you are uninformed, misinformed, microcephalic, or worse."

The protocols we're talking about now, if fully implemented, will allows all manner of wonderful things. Process to process communication between two hosts. Digital Voice. Dynamic routing. But do we need it now? Will we have any process to process applications by 1987? Digital voice? More than one path between LA and Colorado? Maybe by 1990, but not by 1987.

Can we afford to "pay" in 1985 for things we won't need until 1990? In an earlier paper, I postulate no. All we must do is make sure we don't preclude addition of these elements at a later time.

What do we need to do between now and 1987? I submit that we need one thing and one thing only. File transfer.

Direct DX TNC to TNC connections are too expensive for our fledgling network. At best, there are 12 2 meter hops between San Diego and Vancouver. Some of those paths will be marginal at

9600 baud, so let's say 12 hops at 1200, and 15 hops at 9600. Assuming no overhead, that's a rate of 100 baud or 640 baud, depending on the technology used. All the way to Vancouver at 9600 by 1987 is a little iffy. Add in keyup delays, aloha effect, and acks (of whatever type), and we're talking a realtime date rate of less than 20 baud for 1200, and 100 baud for 9600. I think it is safe to say that we won't have a lot of person to person real time chatting going on between San Diego and Vancouver any time soon. Don't disallow it, just don't optimize the network for it over store and forward files.

Are we blessed with several paths to take on our winding way up the coast? Nope. Not by 1987.

Eventually we'll have several choices, 56kb backbone San Diego to San Francisco, 9600 through far northern California and Oregon, maybe 56kb between Seattle/Tacoma and Vancouver. 400 baud through Phase IIIc. 19.2kb through Phase 4A2. But not by 1987.

What will we have? Several centers of activity that want to exchange information in the form of files (much like this one). A fire camp in a canyon that wants to get health and welfare traffic out into the NTS. People using spare capacity to talk real time. Not much digital voice.

And not much bypassing of down nodes on a realtime basis either. We need some reports on downtime figures for digipeaters. MTTR, MTBF, what failed (radio or TNC), etc. It has been our experience here on the lower west coast that when they're up they're up, and when they're down they're down for a long time.

Digipeaters do go down, but the number of state changes is small. Long paths that fade in and out won't be part of a 9600 baud backbone so we need not worry about that by 1987 either.

As I've said before, now and in 1987 there aren't many choices in any case. So discussion of the best failure detection schemes and multiple path routing schemes are premature. As long as we pick a network implementation method that doesn't lock us in.

That's why I like this picture:

Fig 1.

```
Unmodified -(AX.25)--Magic Link---(backbone net----Magic Link -(AX.25)--Unmod
AX.25 TNC             Box            protocol)         Box                TNC
```

Today the magic link box supports fixed routing, because there is only on path anyway. Tomorrow it has externally specifiable routing decision tables. Next make it autonomous. Today it has no provisions for fast forwarding of non-error detected packets. Tomorrow it allows digital voice.

Lyle's NNC design allows us to take the first step, and fill the Magic Link Box slot. It has enough capacity for what we need now, and what we need in 1987. It is expandable. It also has the advantage of several people who have lined up to write code for it.

So that's the kind of network I think we need (in 1987), and the kind I'm working on. What sort of network are the rest of you working on? What attributes does it have, what problems does it solve?

NK6K

C2987 CC229 Howard Goldstein (N2WX,2987) 10/25/85 6:41 PM L:42
KEYS:/RECONNECT PATH BUG/W2VY/

M 7271 Thomas A. Moulton (W2VY,995) 10/25/85 2:14 PM L:37 KEYS:/TNC 2/FADPAD BUG/ TO: N2WX, N2DSY, John

Howie,

I think I found a Bug in the TNC 2 code! (yea yea sure you did, when are you gonna learn how to use it VY???)

I tried to connect to the bbs I use via a poor path 'cause it sounded quiet I then did a double DISCONNECT and re-connected via a better path, I finally got connected, but the path was strange, watch this...

cmd:C WA2SNA-1 V WA2SNA-2 [a couple of tries later] cmd:D cmd:D Retry Count exceeded ***DISCONNECTED*** C WA2SNA-1 V WB2VTN-1 [a few tries] ***CONNECTED TO WA2SNA-1 V WB2VTN-1 ***

Looks good HUH? well just listen to the audio and you will hear that when I SEND it goes Via VTN and when I RECEIVE it goes Via SNA-2 AIN'T THAT WIERD???

The problem is that the paths are not checked to be correct there are 2 options, either verify the entire path to be the same or use the path in the UA connected frame

If the verify of the entire frame fails then send a DISC to via the old path and then when you retry it will make it...

it really sounded funny!

Howie, this ain't too clear, data went from VY to VTN-1 to SNA-1 the ACK went SNA-1 to SNA-2 to VY

lemme know what is still muddy

M 7516 Gwyn Reedy (W1BEL,2975) 10/26/85 7:39 AM L:4
KEYS:/SWEDEN/
TO: (Group 95)

Den Connors, KD2S, visited the Swedish group and has been in contact with them subsequently. Den might be able to provide more information, if you can get hold of him. Gwyn.

C2973 CC34 Jack Brindle (WA4FIB,2963) 10/26/85 2:46 AM L:85

This is an interesting view being re-presented now. It basically says, lets concentrate on the basics and figure out ways to make file transfers more error free. But, let's leave the TNCs out of the networking system. It sounds very interesting, and is close to what many packeteers are calling for. I would ask just one modification. If I happen to have a TNC/Computing system that speaks the same protocol as the "black box" network, then by all means let me tie it in. Maybe not at 56KB (or even T1 rates), but even at 1200 baud it gives me advantages in transferring whatever data we need. The big question we must ask is what will be using the system for, and when. Well, digitized audio is here. It is almost trivial to digitize voice and send it across the wires. Ah, but we have voice radio for that. Packet does provide the unique ability to send audio in a store and forward mode, to be played back whenever the receiving operator desires, however many times he wishes. So what else? How about pictures? MacPacket will be able to send digitized pictures very soon. Sure, initially they must first be saved in a file, then sent as a file transfer. But it won't be long before MacPacket will be able to send the digitized pictures straight from the camera & digitizer. This is happening NOW! Still not convinced? Imagine an interactive game on packet involving many users being quickly updated from a central

station. This type of gaming occurs every day on computers & networks all over the world. An example is the MultiTrek game available from DECUS to run on the PDP-11 and VAX. Why not bring this to packet radio? These things require high speed communications between many users, something that a real network is uniquely positioned to provide. The desire of traffic handling through packet across the nation (one of our major justifications) will not happen with the present system without additions to extend our coverage range. The WORLI BBS is one of these extensions, providing non-real time network messaging services.

Essentially, what I am advocating is a system that allows present TNC users to work side by side with those running more advanced systems. It should handle the present users well, but offer additional services to those of us with advanced hardware and software.

There is an additional consideration in our efforts to build a system. The TNCs and controllers we design are finding their way into the commercial market. Ask AEA where their PKT-1 sales efforts are directed. The commercial market is ripe to literally explode into packeting. What is stopping them? Two things, a viable packet communications system, and government regulations. The second is far easier to handle (yes I know about red tape) than the first. This is why we must carefully design our networks.

Another question worth asking is whether packet is simply a fad. CB, video games and home computers have so far fallen into that category. Could packet be a fad for ham radio? The availability of cheap TNCs parallels similar events in the three "fad" markets previously mentioned. We will undoubtably see a major price war as the vendors try to capture large market shares to make their money quickly before the market goes away. Let us hope not. The good effect on this is while it is happening it gives the system builders extra impetus (and funds) to build the systems. Let us use it well.

Now, for the file transfer functions. Those following my long monologs have noted a new session protocol. It is designed to enhance file transfers and QSOs alike. Combined with a machine-dependent file protocol (in the case of MacPacket, that is MacBinary) it will allow any kind of file to be transferred from one computer to another, and be duplicated on the receiving system in a state identical to the way it existed on the sending system. This means, for example, that MacPacket users can send an application from one Mac to another, then immediately execute the program. It is an extremely versatile protocol. The penalty for using it? about 5 bytes per packet. It's advantages far outweigh that. So when will you see it? Be patient, the Southnet Atlanta Packetfest is only four weeks away.

Now for a plea, directed at Harold, Howie, and anyone else designing TNC software. It has been noted that most TNCs are connected to computers. It is also interesting to note that there are many hams running PCs and Macintoshes, which cost more (but do more) that the Vic 20 and C64. To adequately interface the computer to the TNC, we need two things. The first is a decent hardware connection. We have that in the RS232 wiring. The second is good computer compatible TNC firmware. It is unreasonable to require a computer wade through verbose text to collect information that should require one character. WA8DED has made a good effort at designing such an interface. Please consider adding something similar to your code. An example to follow is the method used to interface microprocessor peripherals to a mpu. The devices generally have a command/status port and a data port. The mpu must specifically ask the device for status or data, it does not just volunteer it to the processor. This interface has proven itself time and again in many environments, not just computers. As I would envision it, the TNC would have two modes of user interface. One very terse, for computers. The other verbose, for humans. Let me provide my own style of verbosity, don't force it on me.

Thanks for listening. We must be doing something right, look at all the attention we have gathered. It continues to amaze me that hams still have the lead in packet radio, far and away ahead of the commercial outfits. I wonder, though, how many commercial concerns have approached TAPR for help in establishing their own commercial networks. My own MacPacket has drawn inquiries for many interesting commercial uses. Like I said previously, the commercial packet world is about to explode. It will be interesting to see where the mode we have pioneered will take us!

73, Jack B.

C2973 CC35 Phil R. Karn (ka9q,2979) 10/26/85 5:57 AM L:59
KEYS:/COMPATIBILITY VS PROGRESS/

Harold, thanks for your comments, they are very well taken. I could take a rather cynical view and say that they prove beyond a shadow of a doubt that amateur packet radio has grown far too fast for its own good. However, if I read between the lines I get the sense that we aren't fundamentally at odds, only that we disagree in our short term priorities.

First of all, we seem to agree that our ultimate goal is a "homogenous" network. Second, we agree that ad-hoc kludges (such as our means of accessing EIES) CAN be made to work in a limited fashion (although if you were accustomed to something that works much better, you might also consider it far from satisfactory). Third, we agree that there are a lot of TNCs out there, (and I might add that people will invent ad-hoc kludges no matter what we do.)

However, at this point I must take issue with some of Harold's remarks. He points out that packet radio is growing rapidly, and that this is an argument for supporting "the unmodified TNC". I would respond that if packet radio is growing this rapidly, then we had better fix things ASAP, because it'll only get a lot harder as time goes on. I know that burning 14,000 sets of EPROMs is a pretty horrifying thought, but aren't these TNCs ever going to have software updates? I can't imagine that the thousands of TAPR TNC owners out there are going to run version 3+.whatever from now until doomsday. Nor do I hope that all the bugs present in other manufacturers' TNCs will remain forever unfixed. And it takes the same amount of effort to burn and distribute ROMs containing a set of minor bug fixes as it does to burn and distribute ROMs with major new features (like support for TCP connections as well as vanilla AX.25).

Amateur packet radio currently suffers from a severe lack of long range planning. The sooner we could get our act together, agree on a network and transport protocol and indicate to the manufacturers and to the general amateur community what our plans are, the sooner will come the day that every TNC will implement it. Yes, "magic boxes" are probably inevitable as backward compatibility aids. But we had better keep our ultimate goal in mind and stress to everyone the temporary nature of the intermediate steps. If we don't, we're liable to end up with "magic boxes" (and probably worse) as permanent warts in our network and don't think any of us want that.

Harold says that amateur packet radio is the "real world", at least in the sense that we don't have a lot of money to waste. Absolutely! THAT'S why it's so important that we look a little farther ahead than our next PC board design and stop wasting our time and money on stopgap solutions. There once was a time that the "bearded experimenter" types could get together and discuss fundamental issues like protocols, addressing and routing techniques and really get some good ideas flowing. Unfortunately, things suddenly became "real" before we knew it (and before we had finished the groundwork discussions), and now many of the "bearded experimenter" types find that they need 30 hour days just to talk at hamfests, write introductory articles for the masses, and (at least supervise) the stuffing of parts into hundreds of boxes for UPS shipment. It's certainly understandable that these people have little time or energy left to think about the relative merits of TCP/IP and X.75. As the story goes, when you're up to your ass in alligators, it's very difficult to remember that your original objective was to drain the swamp.

Phil

C2973 CC36 Phil R. Karn (ka9q,2979) 10/26/85 6:01 AM L:136
KEYS:/KA9Q PROJECTS/

I'd like to describe to this group what my current plans and projects are.

1. I'm currently debugging a TCP implementation written in C. Compiled on an 8088 the object code is a little over 5K bytes. I'd like one of the TAPR software wizards to take a look at it and see what it'd take to add it to the TAPR code, along with an option for initiating a "raw" AX.25 connection or a TCP-on-top-of AX.25 connection? If this works, it might take care of at least a few thousand of those 14,000 TNCs.

2. I've had the BRL IP gateway code sitting here for a month or two. (Note: in Internet parlance, an "IP gateway" is synonymous with a packet switch that switches IP datagrams. This is not to be confused with a "protocol conversion gateway".) This code is in C and was originally intended to run on an LSI-11, a machine with a 64K address space. It speaks the EGP (Exterior Gateway Protocol), the standard routing protocol on the ARPA Internet, and therefore provides automatic routing.

Assuming that the object code size is the same when compiled for a Z-80, I will try to bring this package up on a Xerox 820 (some space can be freed by deleting the drivers for such devices as Ethernets and high speed fiber links). I will attempt to integrate this code with the AX.25 level 2 implementation I've already written, and will add support for the "serial line IP" (SLIP) protocol that allows IP datagrams to be sent over asynch interfaces. With a FADCA card, the result would look like this:

```
 -------------------
|   IP gateway      |
|     code          |
|-------------------|
|  SLIP   |  AX.25  |
 -------------------
     |  |      |  |
     ^  ^      ^  ^
   asynch     synch
```

The IP level packet switch will therefore see four ports: two happen to be synchronous and are of course connected to radio modems. The asynch ports could be used in the following ways:

1. Connected to a local "client" host speaking TCP and running some sort of application, such as a PBBS. Since my TCP is fairly portable, and SLIP requires only an asynchronous interface, this means that a lot of machines could speak end-to-end TCP over packet radio without requiring either an HDLC interface on the machine itself or TCP to be implemented on an external TNC.

2. Connected to a local "client" host programmed to act as a TCP "PAD":

```
                           asynch to
                           dumb term
                               ^
 ----------                    |
|terminal|                     |
|--------|                 ----------
|Cmd Int |                | Cmd Int|
|--------|                |--------|
|  TCP   |       or       |  TCP   |
|--------|                |--------|
|  SLIP  |                |  SLIP  |
 ----------                ----------
     |                         |
     ^                         ^
   asynch                    asynch
     to                        to
```

```
                 IP gw                     IP gw

                  (A)                       (B)
```

"Cmd Int" is Command Interpreter, the thing that you find in every PAD which prompts you for a destination address and does other PAD-like things.

Note that version (A) is already available off-the-shelf, FREE, in the form of the MIT PC/IP package. Besides allowing remote login (i.e., PAD services) across the internet, this package supports reliable file transfers (did I hear somebody say "file transfer" earlier?) with the upper-level TFTP protocol.

Since IP is spoken over the serial lines to the IP switch, there is no need for any form of hardware flow control on the serial lines. This is handled end-to-end by the TCP's, and should the buffers in the 820 occasionally overflow the TCPs will take responsibility for retransmission.

3. Connected to another IP gateway, either local or across a nailed-up phone line. The other gateway could be another 820, providing access to two more radio channels, or it could be a UNIX machine with access to the Internet...

Note that if this works on the 820, it should also work on Lyle's NNC.

Now to the subject of backward compatibility. Since I guess I'm resigned to having an AX.25-to-TCP protocol conversion gateway in here somewhere, (at least temporarily) it could be built on another 820 or NNC, programmed like this:

```
 -------------------
 |      Cmd Int    |
 |--------|--------|
 |  TCP   |        |
 |--------|  AX.25 |
 |  SLIP  |        |
 -------------------
     |        |  |
     ^        ^  ^
   asynch    synch
```

Note that AX.25 here is shown at the same level as TCP, with the "Cmd Int" layer connecting the top of TCP with the top of AX.25. This is the canonical protocol conversion gateway. Logically, it should really be folded outward like this:

```
synch <-------|AX.25|<--|Cmd Int|--->TCP--->SLIP--->asynch
```

This is basically the same thing as my TCP PAD shown earlier, except that the terminal interface on the top is now replaced with an AX.25 module "turned upside down". A user would connect with a dumb AX.25 TNC to the AX.25 side and then converse with the TCP command interpreter. As long as things are kept simple (i.e, no more than one session at a time, and so forth) this will be a doable project -- but it will be limited in functionality, and as I said earlier, meant only as a stopgap.

Another possibility would be to take an ordinary AX.25 TNC and connect it to the serial port of PAD version (B) shown earlier; of course, this could only support one user at a time.

These are just a few examples on how the various software modules I'm writing or obtaining could be fit together. If we had a machine large enough, all these functions (IP packet switch, AX.25 link control, incoming AX.25 "user" ports, TCP "pad" functions, BBS services) could be put in one place. A 68000 can easily do this and in fact 4.2BSD (which runs on 68Ks)

already provides everything here except the lower level AX.25 interfaces -- but X.25 support, both for "dumb PAD" incoming calls and for IP relay, is already there.

Phil

C2973 CC37 Thomas A. Moulton (W2VY,995) 10/26/85 10:23 AM L:5

Harold,
Glad to hear your comments.

Phil,
I think we have.

C2987 CC230 Tom Clark (W3IWI,2976) 10/27/85 12:25 AM L:22
KEYS:/BLACK-HOLE MODE/ONE SOURCE DISCOVERED/INVOLVES CANPAC CHARACTER/

Possible "black-hole mode" (BHM)

A corrupt disk file found one way to create a BHM which drove me up the wall for a couple of days. On the 'IWI BBS, I kept finding the TNC2 croaked each morning. After reviewing the logs, I found that it always occurred when the same file was being downloaded. The file did not cause the TNC1 on the other BBS port to croak. I investigated the file and found that in one spot some garbage had gotten embedded. Buried within that garbage was a ctl-c followed about 20 bytes later by a ctl-y. Reading the TNC2 book, I found that the TNC2 had done just what it was advertised as doing. Ctl-C sent the TNC into cmd: mode, where the CANPAC character (default ctl-y) acted as a toggle to freeze all output from the TNC (chapter 6, page 14, last 2 paragraphs). I don't think that this was the BHM I reported previously, but who knows? It is possible that the over-run condition I described earlier could have overrun the UART creating the Ctl-C followed later by Ctl-Y condition.

Howie, is there any character that really disables CANPAC (like zero)? This looks like a vulnerable mode for RLI BBS's which run in CONV vs. TRANS mode.

73, Tom

C2987 CC231 Howard Goldstein (N2WX,2987) 10/27/85 8:58 AM L:36
KEYS:/BLACK HOLE/CANPAC/CTRL Y/CLOCK/DATA LOSS/
A: 230

Tom -

It's probably a pretty good idea to set CANPAC $00 with the MailBox, and it will really disable the action in both CONV and cmd: modes.

I've reproduced another black hole on filling-buffers, but this mode seems okay (i.e. expected). Viz, after eight packets are queued on an individual stream, no more may be queued until something is ACKed.

Internally this is how information-for-packetizing is handled:

1. data pending packetizing? if so, goto (5)
2. process any ^r/^x etc
3. push the character onto the "packet edit" buffer
4. complete packet built in the buffer? No, EXIT
5. insert the packet into this connection's/link's list.
6. did the packet fit in the list from (5)?
   Yes, flag the edit buffer as empty now and EXIT
   No, Set not rdy flag and next time through, go directly from (1) to (5)

Basically what this says is that in CONV mode, editing functions (like ^X) will not work IN CONV MODE while a packet waits in the edit buffer for queuing.

Note however that echoing, and CMD: line editing are unaffected by this wait.

Other things: CLOCK- I think I found why the clock was performing so poorly. In fact the clock on my 10.0 RLI is now 2 minutes FAST after being set yesterday afternoon. This poor performance was noted quite early on in the multistream software, I think the cause has been corrected.

CTRL-Y performance: Enhanced ^Y handling so it cancels output on the fly. It impacts on the total throughput of the board since its handled in a real time interrupt and it needs to be tested at high speed before its released like this.

DATA LOSS: Skip's data loss problem seems to be related to - hang on now - echoing, and of all things, the clock. More definitive data later, but it appears that when the echo-processing buffer became full things started slowing down. Enough that asserting hardware flow control was delayed past the overflow limit of the input buffer. 73 Howie

How 'bout the TNC sends a length byte before all that is data when some (as yet undetermined) command is set? I think Lyle suggested something like this a while ago... 73 Howie

C2987 CC203 Skip Hansen (WB6YMH,2964) 10/14/85 11:09 PM L:21
KEYS:/MULTICONNECT STRANGITIES/

Howie,

I was testing 1.1.1f with Harold last night and found an interesting characterstic. I was connected to myself thru a digipeater on stream A with user set to 4, when Harold attempted to connect to me he got a busy message and I saw a connect request message. When I switched to stream B and connected to him all worked ok. Streams C & D were in the disconnected state. AX25LVL2 was on and connect mode was transparent.

Another thing I ran into (which was basically an cockpit problem) was when I finished playing with multi-connect and put the host back on line I was active in stream B, I set user to 1 and forgot to reselect stream A. During this time the TNC would set garbarage beacon packets and ignore connection requests. When I reselected stream A all returned to normal. I might be worth while to force the selected stream into the legal range when user is changed, or prevent user from being changed until a legal stream is selected manually.

73's Skip WB6YMH

M 3970 Tom Clark (W3IWI,2976) 10/15/85 11:51 PM L:26
KEYS:/DUAL-PORT DIGIPEATER ON TNC2?/A MODEST PROPOSAL/
TO: (Group 95)

The following was sent to KE3Z @ W1AW via EASTNET and I thought I'd share it with you. John sent back a reply indicating he was interested in the idea and would take a look at it:

\- - - - - - - -

John:

Came up with an idea I though I'd bounce off you re Dual-port digi (DPD). The thought began by noting that (a) X820's are starting to dry up, (b) the multiple power supply is a hassle, (c) you still have to add the state machine and modem for 1200 baud use and (d) the $ and time cost for integrating the 820, power supply, 1200 baud modem, state machine, 9600 baud modem, etc. into a case will deter many from getting DPD's on the air to help with our present congestion problems.

The critical parts of the TNC2 design look very much like an X820 and it has one state machine plus modem included as well as having been designed for lower power consumption. How much trouble would it be to port the X820 DPD code over to the TNC2? The present serial async port would have to have some minor mods (like bringing the clock lines to the DB25) to use the K9NG/KE3Z add-on boards for 9600 but the TNC2 looks like a better DPD "engine" than the 820 to me.

Thoughts?

73, Tom

M 4008 J. Gordon Beattie Jr. (N2DSY,2990) 10/16/85 1:28 AM L:5
KEYS:/KE3Z CODE ON ALL KINDS OF MACHINES !/
A: 3970 TO: (Group 95)

Tom et al:

I have ported the KE3Z code to the Protocom board I mentioned in earlier messages. It works fine. I can't see any major problem with porting Jon's

code to the TNC2. 73, Gordon

M 4022 Lyle Johnson (WA7GXD,2973) 10/16/85 2:17 AM L:11
KEYS:/TNC 2 CABINET END PANELS ARE ENROUTE/
TO: WA7GXD, (Group 95)

Just in case you were wondering, the cabinet end plates came (finally!) and this time they actually fit !!!!

If you are among the lucky 511 folks who were sent a TNC 2 kit earlier you will be happy to note that the cabinet end plates were sent out to you by first class mail today (overseas by airmail).

With luck, the next lot of 100 to 300 kits should go out sometime next week and we should be caught up by the end of the following week.

Hang in there! TNC 2s are coming!

C2987 CC204 Howard Goldstein (N2WX,2987) 10/15/85 11:00 AM L:3
KEYS:/MULTICONNECT STRANGENESS/RELEASES 1.1.1A-F/
A: 203

The strangeness in USERS > 1 is corrected in the 16K RAM 1.1.1 (cksum $2F)

Releases 1.1.1a-f act like USER 1 when USER > 1. 73 Howie

C2987 CC205 Tom Clark (W3IWI,2976) 10/15/85 11:31 PM L:17
KEYS:/1.1.1G RUNNING AT IWI/MORE ON DATA "BLACK HOLE"/

Howie, got 1.1.1g (16k version) up and running in the Rev.2 board here with no problems (it is the only one I have with 16k ram; put code into a 27128). Did a multi-connect test and all seemed OK.

With 1.1.1f I again got into a "black hole" situation. In this case I apparently had FRACK set too short and TNC's async buffer over-ran without sending an X-OFF. After the buffer flushed itself on the air, data from computer (terminal) to TNC was apparently getting to TNC and going out, but there was no echo from TNC and no data from TNC to terminal. It almost looks like TX buffer is overwriting some crucial part of RX code. The only way to clear the fault was to power down. In our user notes we MUST specify that FR must be big enuff to take the outgoing packet; rule of thumb I used is that FR should be numericaly > or at least = MAX. I have had this "black hole" report from a couple of local users too. Ideas?

73, Tom

C2987 CC206 Lyle Johnson (WA7GXD,2973) 10/16/85 2:09 AM L:26
KEYS:/BEACON BUG?/INFO NEEDED ASAP/ONWARD!/

Howie,

One of the local beta testers, Eric, N7CL, notes the following on 1.1.0 code.

Beacon After N seems to reset itself after it sends the beacon and thus turns into s beacon every N. Highly undesireable. I also must note that Eric runs with no squelch so the DCD line will have some chatter on it. I may have missed an earlier comment that deals with this.

Also, we are about to print up the errata sheets/software notes for the next (last?) batch of TNC 2s. Any details you can give me on any of the commands not found in 1.1.0 will be very useful -- especially the treatrment of streamswitch when users = 1 and users <> 1 and the impact

on transparent mode. YUou have seen the earlier systems manual, so you know the kind of info I need. We hope to send out a hundred or two kits by early next week with 1.1.1 EPROMs (I spent tonite writing a pascal filter to take the offset files in the 1.1.1 hex dump and offset them back to the correct address -- the eprom programmer we bought doesn't support an offswet address!!!! I needed the experience, it workls fine noew, I had nothing else to do anyway...)

Cheers!

Lyle

M 4082 Dave Pedersen (N7BHC,2960) 10/16/85 9:07 AM L:21
KEYS:/TNC-2 PROBLEM?/
TO: (Group 95)

One of our group here in Salt Lake has discovered what may be a bug in the TNC-2. I have duplicated the problem on my TNC-2. Is it a problem on the TNC, or just pilot-error?

FROM: Steve Baxter KA7JXR
RE: TNC-2 Problem ?

A few days ago, I decided to experiment with the 8bitconv command on the TAPR TNC-2. When I turned it ON, the TNC appeared to develop problems; it would not respond to any command except control-X, thus I could not get out of the 8bitconv mode, or do anything else either. The only solution I found was to unplug the RAM chip.

Is this a problem on the TNC-2, or am I just not doing something right?

73...Steve KA7JXR

M 4255 Howard Goldstein (N2WX,2987) 10/16/85 6:18 PM L:11
KEYS:/TNC-2 PROBLEM FIX/
A: 4082 TO: N7BHC, (Group 95)

Dave,Steve,

I'm sorry confirm what you discovered already; 8 bits were also being looked at by the TNC in cmd: mode, belying the CONV part of 8BITCONV. This will be corrected in the next distribution release!

If you want to run 8BITCONV on with your release, make sure AWLEN is set to 8 and your terminal/computer is set up to handle 8 bits. Provided commands and parameters are issued with the 7th bit low, it should work okay until the new s/w comes out.

Thanks for the report!

73 Howie

(Modification:)
C2987 CC200 Howard Goldstein (N2WX,2987) 10/16/85 6:04 PM L:0
(ORIG.) 10/14/85 7:47 PM L:38
KEYS:/WAS 1.1.1(G)/UPDATED IN CC207/

C2987 CC207 Howard Goldstein (N2WX,2987) 10/16/85 6:03 PM L:58
KEYS:/HERE WE GO AGAIN/+READ FOR 1.1.1(H)/CKSUM $7F/REQUIRES 16K RAM/

To: Beta
Fm: Howie
Re: 16K RAM version 1.1.1h, ten connection, release description
Dist: Closed, for now

The .HEX of the release described hr can be had with +READ @(thisitem).

Release 1.1.1(h) - checksum $7f - follows on the heels of (g) pretty quickly and fixes (again) something, changes (again) something, and adds (again) a few things:

o - STREAMCa ON|OFF default: OFF. When ON, the callsign (if connected) of the other TNC is displayed between colons immediately after a TNC originated stream switch. Ex:

```
|A:K4NTA-2:hi howie
hello ted how goes it?
|B:WA7GXD:*** CONNECTED to WA7GXD
|Bmust be a dx record. ge lyle
|Aunreal ted! fl-az no digis!
|B:WA7GXD:big band opening...ge
     etc.
```

What used to be like "|A" is now like "|A:KV7B:" if the parameter is ON.

o - STREAMDbl ON|OFF default OFF - This setting, when ON, will send all STREAMSwitch characters, except those that result from real STREAMSwitches originated by your TNC. STREAMSw must NOT be one of the physical stream letters used by your TNC for this to operate correctly.

o - ** DELETED - TRAPSTREam ON|OFF (replaced by STREAMDbl)

o - ** FIXED - 8BITCONV affecting commands problem

o - ** FIXED - version 2.0 protocol problems

o - ** FIXED - Errant Beacon AFTER startup problem

----- Below are earlier 1.1.1(g - checksum $2f) changes

TNC 2 software for 16K RAM is done. Buffer enlargement is one of the things addressed. Here are some other changes:

o - Up to TEN connections/links are allowed

o - A clock adjustment is installed. The associated command is CLKadj nnn, where 0=<nn=<65535. The adjustment effects the time by subtracting 9.1667ms every 100nn msec. Ex:

only WHEN CLKadj != 0,

$$\text{speed change of clock, in \%} = 100 - 9.16667 * \frac{1}{\text{CLKadj}}$$

o - *** DISCONNECT messages are stamped according to CONstamp

o - USER command is changed to USERS, and

o - The USERS parameter has new significance:

```
USERS = 0  allows incoming connections on any free stream
USERS = 1  allows incoming connections on stream A only
USERS = 2    "  "      "  "     "       "  on streams A & B
USERS = 3    "  "      "  "     "       "  on streams A,B, & C
        etc...
```

o - A bug where the output stream didn't change on ^C from CONV mode on a different stream has been fixed.

73 Howie

M 4486 Thomas A. Moulton (W2VY,995) 10/17/85 11:59 AM L:8
KEYS:/TAPR UPDATE/
TO: (Group 95)

Here is your TNC 2 Update for Wednesday October 16

The cabinet end panels have been mailed for up to order number 511

The TNC 2's will be shipped starting the week of Oct 28
This will be starting with order number 512

Orders are still being taken for the November time frame

C2987 CC208 Howard Goldstein (N2WX,2987) 10/17/85 4:37 PM L:94
KEYS:/SOFTWARE NOTES/COVERS 1.1.1 CKSUM $7F RELEASE THROUGH 1.1.0/
A: 207

software notes - version 1.1.1 - 17oct85

o - The signon checksum is $7F

o - STREAMCa ON|OFF default: OFF. When ON, the callsign (if connected) of the other TNC is displayed between colons immediately after a TNC originated stream switch. Ex:

```
 |A:K4NTA-2:hi howie
 hello ted how goes it?
 |B:WA7GXD:*** CONNECTED to WA7GXD
 |Bmust be a dx record. ge lyle
 |Aunreal ted! fl-az no digis!
 |B:WA7GXD:big band opening...ge
     etc.
```

What used to be like "|A" is now like "|A:KV7B:" if the parameter is ON.

o - STREAMDbl ON|OFF default OFF - This setting, when ON, will send all STREAMSwitch characters, except those that result from real STREAMSwitches originated by your TNC. STREAMSw must NOT be one of the physical stream letters used by your TNC for this to operate correctly.

o - CLKADJ nnnn default 0 - zero is no adjustment
only WHEN CLKadj != 0,

$$\text{relative clock spd. , in \%} = 100 - 9.16667 * \frac{1}{\text{CLKadj}}$$

o - USers n default 1 - The parameter may take on values from zero through 10. USers affects ONLY how incoming connect requests are handles, and has no affect on the number or handling of connections you may initiate on your own.

```
USERS = 0  allows incoming connections on any free stream
USERS = 1  allows incoming connections on stream A only
USERS = 2   "   "      "   "     "        "   on streams A & B
USERS = 3   "   "      "   "     "        "   on streams A,B, & C
     etc...
```

o - TRIes n 0<=n<=15 default ? - This command is used to retrieve and set the count of tries on the currently selected input stream.

If RETRY is set to 0, the value returned by issuing a TRIES command will always be 0. Use of this command to force a new count of tries is NOT recommended.

o - STReamsw n 0<=n<=255 default $7C "|" - The STReamsw command is used to select the character used by both the TNC and the user to indicate that a new stream - that is, a connection - is being addressed.

The character can be PASSed in conv mode. It is ALWAYS ignored as a user-initiated stream switch in transparant mode, and flows through as data. This means that the outgoing stream can not be changed while "on line" in transparant mode. <<see further STREAMDb and STREAMCa commands>>

o - CStatus - This is an immediate command

which shows the stream identifier and link state of all ten streams (links), the current input and output streams, and whether or not each stream is "Permanant" (see CONPerm). Ex:

```
cmd:CS
A stream - IO Link state is: CONNECTED to 305MLB
B stream -    Link state is: CONNECTED to AD7I P
C stream -    Link state is: DISCONNECTED
D stream -    Link state is: CONNECTED to WA7GXD via K9NG-2
            ...
I stream -    Link state is: CONNECT in progress
J stream -    Link state is: CONNECTED to KV7D via KV7B-1
```

The example above shows the A stream is assigned both the input and output streams and the B stream is connected to AD7I "Permanantly". All the other streams' states are shown as they'd normally appear when a CONNECT command without parameters was issued.

o - RESTART - This is an immediate command which has the same effect on the TNC as shutting it off and turning it back on again.

o - MYAlias ccccc Default: blank - The parameter of this command is an amateur callsign-ssid that this TNC recognizes for digipeating purposes ONLY.

In some areas, wide coverage digipeater operators have changed the callsign of their machine to a shorter and (usually) easier to remember identifiers like ICAO airport id's or other well known abbreviations.

Use of this command permits HID to identify normally with the MYCALL callsign yet permit an alternate "alias" repeat-only call.

-*-*-*-*-*-*

There are a few operational changes in this release that should be noted:

- All connect requests from stations with totally blank callsigns are rejected with a busy response.

- The *** DISCONNECT prompt is daytime stamped according to the CONSTamp setting

- TNC 1-like transparant mode is supported only when one connection is established and USERS is set to 1

-*-*-*-*-*-*

- in the TNC 2 System Manual, chapter 6 p 37 = MHeard command, change the line that reads:
Stations that are heard digipeating are marked with a * in the heard log.
to
Stations that are heard through digipeaters are marked with a * in the heard log.

C2987 CC209 Tom Clark (W3IWI,2976) 10/17/85 10:06 PM L:13
KEYS:/HOWIE: "BLACK HOLE" BUG?/LYLE: HEX => BIN CONVERSION FOR PC/

Howie: were you able to do anything about the "black hole" bug where the TNC2 refuses to echo anything when it gets an input buffer overflow with FRACK too short to support the outgoing packets? I've now had 6 local users report this one to me. My standard fix is to tell them to increase FRACK and to pull the RAM to achieve a reset from the bogey condition.

Lyle: I have a PC program that converts Intel HEX into Binary. It is written in BASICA and runs quite fast compiled under BASCOM. It does handle offsets and also insures that all trailing (unused) bytes in ROM

are defaulted to $FF. Need a copy?

73, Tom

C2987 CC210 Howard Goldstein (N2WX,2987) 10/17/85 11:06 PM L:11
KEYS:/BLACK HOLE/
A: 209

Tom -

I'm going bonkers trying to reproduce this one - really!

I set up for AW 8 PAR 0, leave XFLOW/XON/XOFF at default, set MAXFRAME 7 and FRACK 1. Then connect via 2 digis and open er up but still no crash. Note my data source ignores flow control from the TNC for this test and keeps sending the whole file, overflowing every char of the way.

Other commands left at defaults.

Will try gorging it interfaced a few different ways tomorrow and see what happens.

73 Howie

M 4853 Dave Pedersen (N7BHC,2960) 10/18/85 8:51 AM L:12
KEYS:/56KB MODEM/DIGITIAL AUDIO/
TO: (Group 95)

Bill Gillman, WA7RFS, in Salt Lake City, is interested in experimenting with digital audio transmission, and poses the following questions:

Is anyone else playing around in this field?
What developmental work is being done with a 56kb modem?
Has anyone used the TNCs at 56kb?
Where can I find information on the modem used in Swedish Softnet system?

I hope someone can be of help to Bill.

73...Dave N7BHC

M 5202 Steve Goode (K9NG,2981) 10/19/85 12:11 PM L:19
KEYS:/56KB MODEM?/
A: 4853 TO: (Group 95), K9NG

Anyone responding to Dave Pedersen's request for info on the Swedish Softnet system please send a copy to me also. I have heard rumors of this network for about a year but have never received any solid info on it. I have received requests from Germany on my 9600 bps modem and wonder why they would be interested in such a slow date rate (compared to 56 kbps) if the 56 kbps is really available. Any solid info would be greatly appreciated here.

I know of there groups that are interested in higher data rates than 9600. They all have my modems going. The problem is that the people interested in higher rates are also the people installing the 9600 bps backbones (as I see it) so until the first backbones are in place and these people are free for new experiments it will be difficult to get higher rates.

At this time any higher rates would probably be scaled versions of the 9600 bps modem requiring wider receiver IFs. Anyone with other ideas please send them along.
73
Steve

M 5300 Pete Eaton (WB9FLW,2970) 10/20/85 12:55 AM L:38
KEYS:/TNC 2 BLUES/BUT THE CARDS WILL WIN THE SERIES/

TO: (Group 95), WB9FLW

To: All
From: Pete WB9FLW
Subject: How NOT to put your TNC 2 rev 2 board together!
Dist: Open (but...NO smart a-- replies allowed!)

This evening Mel Whitten K0PFX and I were finishing off my TNC 2 rev 2 board, this revised circuit board requires a small piece of coax that acts as a faraday shield from the dc power plug to the 7805 regular. While doing this simple proceedure we were watching the St. Louis Cards destroy the K.C. Blues.

Since a child of 10 could solder this coax to the board we spent our mental energy on the game. When we turn the TNC 2 on to smoke test it the results were positive, it did smoke. Seems that for some reason we had bypassed the 7805 5 volt regulator and applied plus 15 volts to the TNC 2's 5 volt line.

This does not work.

Nor does the TNC 2 work, at all. Now all the lights on the front light....all the time (makes a great night light) but besides that nothing.

I hope that TAPR places in their Erratta sheet "do not watch the world series and build this kit" or "please allow your 10 year old to construct this unit if you must watch sports programs at the same time."

Well now that I have alot more free time I can watch the series in the knowledge that my hobby will not interfere...at all.
P.S. the series will not go to 7 games the cards will stomp the Blues BEFORE game six.
P.P.S. don't pay any attention to someones predictions who can put a silly TNC kit together.
PLAY BALL!

C2973 CC4 Jack Brindle (WA4FIB,2963) 10/18/85   7:51 PM L:80
KEYS:/NETWORK DATAGRAMS VCS PVCS/

Phil; But what is your mailing address???
One of the things that really bothers me about the datagram approach is the necessity to contain routing information in every packet, and the resulting length of the packet. I agree that at present we have datagrams in layer 2, and that someday we should get rid of that. (prrobably it will happen about the time that the PID becomes useful - no present TNC code that I have seen allows access to the PID!) The biggest reason to get rid of the relay fields in L2 is that it just doesn't belong there. It served a purpose, but L3 is supposed to eliminate that. Yet it may continue to serve a purpose once L3 is installed.
The name of the game in radio packeting is to maintain short transmissions so that the transmitted data does not get "clobbered" by QRN and fading. The implimentations of datagrams that I have seen assume a very good transmission media, and place extra info in the routing header to lengthen the packet. Of course when one is transferring 4096 byte packets over coax, who cares. Rather than hit you with more stuff that I'm sure you have already thought of, it's probably best to let you explains things first. As soon as I get the TCP/IP info from you (your address, please...) I'll have a better shot at asking intelligent questions.
As a networking person, I too have some ideas on how to impliment the sye system. Feel free to blow these away...
I have problems with the overhead caused by the virtual call. In a limited situation, it is probably ok, but we will fill the airwaves with a lot of packets just establishing the layer 2,4, and 5 connections, not to mention lots of L3 requests inside the network. In a limited fashion,

virtual calls do have their place. Lets assume a virtual circuit system. The VVirtual call should then be used to make connections with nodes that for some reason are not known to the network. This also suggests that some mechanism must exist to alleviate the VC. The answer lies in between the virtual call and the datagram. It is, of course, the Permanent Virtual Circuit. The PVC uses established routing paths well known to all nodes, so that circuits do not have to be built, simply used. The biggest problem with PVCs is the necessity to maintain routing tables in each node. This would cause a "Control" node to be designed into the system to oversee the network and build new tables for each node as new nodes appear. It allows a close-knit system that can be monitored for problems. It also would allow network security to assure that unwanted stations do not invade the sanctity of the system by attaching themselves where they do not belong.

This brings up another question, though. Just where does the network and transport layer code belong? I believe that some form of transport code belongs in the TNC, since the end TNC's are the only communications entities that can verify data on an end-to-end basis. However, we should take a look at making the network look like a black box to the end users, much like telenet does for this connection. The advantages become numerous since the TNC can remain simple, like the Bell 103 & 212 modems we use with telenet (although those have gotten awfully complex). All information necessary to packet routing then is placed in the network nodes, and the end user's TNC can continue to think he is in a point to point connection. The biggest advantage is that the current TNCs can be used with very little modification. This is a major consideration when you look at the various TNCs on the market and the fact that few of them adhere to the current AX.25 standard. This is slowly being fixed as the TNC2 comes out, along with TAPR 4.0 or the WA8DED code for the TNC1. Our biggest problem, then will be the "popular" Kantronics unit, which is definitely non-standard with no hopes in the future of becoming standard (in fact, just the opposite - apparently they are trying to create their own "standard"). It will be a tremendous task to upgrade all these TNCs to use the "new" layer 3 code, whatever it is. Let's face it, packet has been taken away from those of us who like to experiment and build, and given to those who like to buy and "operate."

These are definitely thoughts to consider. I don't necessarily agree with a lot of them, being a network "purist" I'd like to see a lot of thinds done differently, but like the repeater fields, they are here and we might have to accept them.

It seems that PVCs have not yet been proposed to packet radio in general. Please give them some thought, then tell us why you think they will or won't work. Implimented properly, they could give the best of both worlds, VCs and datagrams, at an advantageous cost! One thing I would see in them is that regional networks would be necessary because of the large job of building nationwide routing tables! Oh there is another advantage to having a central controller. It would make software distribution and upgrades easier. Each node would be outfitted with a "baseline" program that would allow the node to boot itself into the network. Then the central node could maintain the latest code release in each node, simply by downloading the new code to the node. No more mailing disks & EPROMS!

Anyway, much to consider. My Session layer protocol is moving along also. I expect to release it at the Atlanta Packetfest. It is extremely interesting and worth a peek. I even hope to have it running in MacPacket by then!

More later. 73, Jack B.

C2973 CC5 Phil R. Karn (ka9q,2979) 10/19/85 2:57 AM L:237
KEYS:/NETWORKING MODELS/

Jack, thanks for your interesting comments.

The overhead of a protocol is a function of both the size of its header and the amount of data the user sends in each packet. More specifically:

efficiency = data/(header+data).

If the header size is increased, it is possible to retain the same overall efficiency by sending more user data in each packet. It is instructive to compare the real numbers used in some actual networks, namely Telenet (X.25) and the ARPA Internet (TCP/IP).

X.25 levels two and three have a combined header of 9 bytes (the 6 bytes in layer 2 includes the closing flag and CRC). The maximum packet size on Telenet is 128 bytes, and we haven't yet figured out a way to negotiate it to a higher value. TCP/IP has a combined header of 40 bytes, but every Internet host is prepared to receive datagrams of at least 576 bytes (512 bytes data plus TCP/IP plus room for options.) Let's compute efficiencies, assuming maximum packet sizes as would be the case during a file transfer:

Telenet/X.25: 128/(128+9) = 93.4%
Internet/IP/TCP: 512/(512+40) = 92.7%

Hardly much different. In fact, many hosts, including ours, have TCPs which negotiate the maximum packet size limit up to 1K, since memory is cheap:

1024/(1024+40) = 96.2%

This is better than Telenet! One almost wonders how serious some of the X.25 advocates really are about header overhead when you look at some real networks. Of course, this is a slightly unfair comparison, since TCP provides end-to-end guarantees and X.25 doesn't. We should add to the Telenet case at least the overhead of a "lean and mean" transport protocol designed for operation over a virtual circuit subnet (e.g., TP-1), and this can only make the Internet look better.

A valid concern, of course, is that the user may not generate large enough packets to overcome the handicap of the larger datagram headers. This was a very real problem with early implementations of TCP when used for remote login over very slow network paths. The Internet presentation-level protocol for remote login, "Telnet", (not to be confused with TelEnet) is often "negotiated" into character-at-a-time mode in order to simplify the use of screen editors. The sender's TCP, since it usually faces a generous window of 1-2K bytes on the receiving machine, would gladly generate a 1-byte TCP segment (TCP packets are called "segments") each time the user hit a key, even if the network was already clogged with lots of little segments still in flight. This became known as the "small packet problem".

A solution for the small packet problem was found by John Nagle at Ford Aerospace. It is elegant and effective enough to now be universally implemented. Basically, it imposes a "one segment at a time" restriction on the sender. If the user hits a key when the connection is idle, a single character segment is sent. However, any further characters are held in a queue at the sender until an acknowledgement is received for that character. When the acknowledgement does arrive, the pending characters are sent out in one large segment, and the process repeats. Only when there is already enough data pending in the transmission queue to generate a maximum-sized segment will more than one segment be allowed to exist in the network at one time.

A little thought will show that this scheme is highly effective at reducing the number of small packets in the network and the possibility of network congestion, and does so with little effect on user-perceived performance. It is a superior replacement for the packet closeout timers often found in X.25 pads, and is easier to implement, too. Still, it is clear that the best solution for interactive users is to find a way to avoid altogether the need for character-at-a-time transmission, but when one is talking to an application that doesn't always operate in terms of "lines", this isn't easy.

I think the trend will be towards smart "network terminals" that understand something about the specific application and can offload many of the functions now done on the hosts. When this is done, "terminal to "host" traffic will look more and more like "host to host" communication: fewer but

larger packets, and a need for more flexible networking services such as those provided in the Internet. This is perhaps THE sharpest contrast between the CCITT and ARPA "universes". The 1970's model of a "dumb terminal" talking through a PAD to a big IBM mainframe thoroughly permeates the design of X.25, and since this is exactly what 99% of X.25 users still do, it's no real surprise that it's optimized for this task. But it's hardly a general purpose packet network, and the growth of personal workstations and local area networks (which are invariably based on datagrams, e.g., Ethernet and PROnet) are already forcing the public data net people to reconsider the services they provide.

You make a valid point about the increased noise vulnerability of large packets. To a certain extent I think that we will be forced to improve the BER of our links anyway just to get reasonable performance, but that begs your question. There are two ways that large datagrams might be broken down into smaller noise targets. One can "internally fragment" each datagram as it is sent over a Level 2 link. For example, AX.25 might forward a single 1K-byte IP datagram as a sequence of four 256-byte I-frames, and the receiving layer 2 would put these back together before passing the datagram back up to the layer 3 module for further processing. Clearly only the first I-frame need carry the IP datagram header, but the link layer must provide sequenced delivery of each piece of the datagram (AX.25 does) with some indication of the beginning and end of each multi-frame datagram (AX.25 doesn't, because this is the function of the M-bit in the Packet Layer of X.25, and AX.25 doesn't have this layer.)

So that it can run on a pure datagram network (e.g., Ethernet, or AX.25 in the disconnected mode with UI frames) IP provides another option, namely "external fragmentation", where a datagram can be broken up at the IP level into IP fragments. They propagate through the Internetwork as though they were separate datagrams and are reassembled only when the reach their final destination. Of course, this duplicates the IP header on each packet and increases overhead, especially since other nodes see extra packet loading as a result. In practice, IP-level fragmentation is reasonable as long as the fragments aren't too small, while link-level fragmentation is best when one link has an unusually small packet size limit. Both methods can and do coexist on the Internet.

You mentioned at the beginning of your note that datagrams "contain routing information in every packet". This is not necessarily so. At a minimum, they must carry only the source and destination addresses in each packet; in IP, addresses are 4 bytes so this is 8 bytes of overhead (included in the 20 byte IP header). Even AX.25 without digipeaters will still require source and destination callsigns, so it remains a true datagram protocol at the lowest level. (In fact, I challenge anyone to give me an example of a protocol operating on shared, multiaccess media that ISN'T datagram-oriented at its lowest level.) Providing routing info in each packet is called "source routing", and AX.25 with digipeaters is an example of this. IP also allows source routing, but unlike AX.25 it is OPTIONAL, and it is not necessary (or even common) that source routes appear in each datagram. Ordinarily, routing tables at each packet switch are consulted to determine the best "next hop" toward the datagram's destination; source routing allows senders to override the routing tables when desired (e.g., to perform network testing or to bypass a routing problem.)

You make some interesting comments about permanent virtual circuits. It has occurred to me that datagrams and switched virtual circuits are merely two points on a much larger, continuous spectrum of network design:

```
Fully                                              Fully
Static                                             Dynamic
Bandwidth                                          Bandwidth
Allocation                                         Allocation

  |----------------|---------------|----------------|
Dedicated        Circuit          Virtual          Datagrams
```

```
Lines                Switching          Circuits

<------ Increasing initial connectivity/setup delay
        Increasing per-transmission overhead ------>

        Increasing suitability for bursty traffic-->
<------ Increasing suitability for predictable, steady traffic
```

Now what do you mean by a "permanent virtual circuit" (PVC), and where would it lie along this line? Does an implicit PVC always exist between every pair of nodes in a network? If so, the "PVC identifier" has to be just as large as the address field. This negates the only selling point of a VC protocol, namely not having to send big addresses in each packet, and your PVC is really the same as a datagram service (except for side issues like guaranteed delivery and sequencing, which may or may not be associated with any particular datagram network).

On the other hand, if PVCs exist only among a subset of nodes, i.e., like a limited collection of dedicated lines, how does a pair of nodes without a direct PVC communicate? Either there has to be a simple, automatic mechanism to create a PVC (which means that they're no longer "permanent"), or there needs to be ANOTHER layer of routing and switching on top of the PVCs in order to provide full connectivity to the higher layers. So I don't really see how PVCs are relevant to amateur packet radio. They are a special service provided to commercial entities that aren't interested in communicating with more than a few well-specified places, but don't have the traffic and/or money to justify leased lines.

Regarding the placement of code: this should be fairly straightforward from the ISO and ARPA reference models. The transport layer must be end-to-end, otherwise it cannot guarantee end-to-end data integrity (which is its major function). In other words, it resides in the client host (TNC or whatever) at each end of a logical "connection", but not in the intermediate packet switches ("transit nodes"). The host must have enough understanding of the network layer in order to exchange traffic with the nearest packet switch, but it need not concern itself with switching and routing (unless it also serves as a packet switch, of course.) This is absolutely trivial with IP, because "speaking IP" requires only that you stick the appropriate header in front of each TCP (or other upper level protocol) packet and send it at your nearby packet switch. The packet switches need not concern themselves with TCP (unless they are also capable of being hosts), but instead must be able to do the things that packet switches do: handle multiple links and route packets between them.

Some people have proposed a networking model that looks like this:

```
Unmodified -(AX.25)--Magic Link---(backbone net----Magic Link -(AX.25)--Unmod
AX.25 TNC            Box           protocol)        Box                  TNC
```

The "Magic Link Box" must now act as a "protocol conversion gateway" (a.k.a. a "transport level bridge"), splicing an AX.25 connection on one side to the "backbone net protocol" (e.g., TCP) on the other. Such Magic Link Boxes have been drawn many times on viewgraphs and on slick ad copy, and at first they look really nice because nobody has to change their existing equipment.

----> HOWEVER, THIS APPROACH IS SIMPLY WRONG!!!! <-------

It is only when somebody actually tries to build or use such a box that they realize how flawed the model is. The first (of many) problem(s) is that there is no longer any end-to-end protocol; TCP can only guarantee delivery between the Magic Link Boxes, not between the TNCs on each end. Other problems: How does the user tell the Box who he wants to connect to? How does the Box tell whether what's coming in on the AX.25 link is to be relayed as data or acted on as a command? Suppose the user wants to maintain multiple connections; how does he indicate multiplexing of his data streams? Suppose the Magic Link Box he's connected to gets hit by lightning; how can he switch

over to another Box in his area?

And so on. By the time you've solved these problems, you finally realize that you've invented a full-blown but utterly ad-hoc "protocol" for commanding the Box over the AX.25 link. Whether you've added this "protocol" to the TNCs (so they are no longer "unmodified") or forced the user and/or his host system to learn it (more likely), you will have discovered that this was far more painful and kludgey than it would have been to do it right in the first place by putting the transport protocol in the TNC or other host where it belongs. The Boxes can then just switch packets, and that's certainly enough to keep them from getting bored (especially Z-80s, since it has been arbitrarily decreed that we shall use them here!)

I put Protocol Conversion Gateways in the same class as Star Wars. Both are superficially alluring. On closer examination, however, they turn out to be extremely difficult, expensive, dangerous and ultimately futile attempts to devise "technological fixes" for what are fundamentally political problems. TANSTAAFL!

One last subject: the session layer. I've never been able to get a clear picture of just what it is that the session layer does, and perhaps this is because its function seems to be unnecessary in any network save SNA. For example, TCP has source and destination port multiplexing, so that one TCP module in a host can handle all of the network traffic from a number of user processes. Depending on your point of view, this means that TCP combines levels 4 and 5 into a single layer, or it makes level 5 unnecessary.

Sorry this has gone on so long, but once you get me going...

73, Phil

C2987 CC211 Lyle Johnson (WA7GXD,2973) 10/19/85 4:25 PM L:15

Howie,

A couple of ?bugs may be in the 1.1.1 release.
1) ctrl-Y (canpac) doesn't work to cancel dispaly output as documented in the system manual chap 6 page 14. I think you need to ctrl-s the data flow, then ctrl-y works...
2) whewn using a 2-wire rs232 hokup (+ gnd) the tnc 2 refuses to send data to the user port unless the CON led is illuminated. Sign on message terminates when the two CON and STA leds extinguish, althoiught e TNC accepts date, it won't echo it until the CON LED is again illumkinated...
3) if you are connected to a station, then force a disconnect by enetering the DISC command twice in rapid succession, the CON led operates backwards 9off when connected, on when disconnected).

Lyle

C2987 CC212 Howard Goldstein (N2WX,2987) 10/19/85 7:50 PM L:10
KEYS:/LEFT OUT EARLIER COMPLEAT GUIDE OF 1.1.1 COMMANDS/RECONNECT/BUGS/

Lyle - will check out bugs in prev comment.

I left out a command from the earlier "compleat" guide of new commands:

RECOnnec call1 [Via call2[,...]]

The parameter call1 MUST match the callsign of the station already connected to on the current input stream. Other parms as per required for path.

It's illegal to either: issue this command while link state is: DISONNECTED (and a '?not while disconnected' message appears) or change the call1 parameter from the current station.

73 Howie

C2987 CC213 Skip Hansen (WB6YMH,2964) 10/19/85 8:08 PM L:28
KEYS:/HARDWARE FLOW CONTROL ?/

Howie,
Got another one for you. I was sending a file to another PBBS today using hardware flow control (stop=start=xon=xoff=0, rflow=xflow=off awlen=8,parity=0,autolf=on,8bitconv=off,nulls=0) in converse mode and data was getting dropped badly. I thought it was the PBBS until I tested with Harold and found that with exactly the same configuration in place the file would transfer perfectly in transparent mode, but drop large hunks (packets ?) when in converse mode. We tried with echo on and off, with no difference. The hardware flow control is going active and my computer is stopping, the echo from the TNC is perfect on the screen. The dropped data doesn't not seem to occur anywhere near where the TNC flow controlled the incomming stream. I also tried maxframe=1 and paclen=60 with the same results. The only way I was able to get the data thru converse mode was by sending 16 characters then waiting for about 1 second, then sending another 16, etc. The channel was very uncongested (4:15 am) at the time of the test. I also tried direct as well as thru 2 digipeats with no difference.

I just ordered the 8K X 8's and 27128's to upgrade Tiny with, but I may end up stealing the RAM from my TNC-1 if I get impatient.

In case you are woundering why all of the activity from me of late I have been on vacation for 2 1/2 weeks, but fear not, I am going back to work next week! Thanks for the fast response on my stream switch suggestion!

73's Skip WB6YMH

C2987 CC214 Tom Clark (W3IWI,2976) 10/20/85 12:56 AM L:44
KEYS:/WORLI BBS COMPATIBILITY/ALSO: VERSION 2 COMPATIBILITY TEST RESULTS/

Howie, there have been several EASTNET msgs going back and forth between WORLI and me about handshaking/over-run problems with TNC2. The main problem now seems to be in the monitor mode, with some characters being dropped. Apparently Hank handles the handshaking with the TNC2 in this mode using FLOW -- when he wants the TNC to go mute he sends a <space> and to get it to resume he sends a <bs>. Apparently this works fine with the TNC1, but TNC2 usually gets a couple of extra characters in there so that the <bs> doesn't obliterate the <sp>, but rather some other characters. I manually simulated this on the BBS 820 in terminal emulator mode. If XF ON and if xon/xoff characters are used, the problem seems to go away, so I have suggested to Hank that he try more conventional xon/xoff handshaking. More details later after I get an answer from Hank.

I have now received RLI 10.2 BBS which now recognizes all the new monitor mode outputs and properly logs activity under the "J" command. Let me know if you want a clone.

All: We ran an interesting AX25 Version 2 compatibility test here this weekend. I put the BBS on with AX ON to see if we would have any problems. The news is good.

As was expected, Version 1 users connecting to the BBS caused the TNC2 to revert to Version 1 protocols, so the change was transparent to them. Those users running TNC2's or WA8DED TNC1's with Version 2 found that everything worked fine for them too, except that (as we knew would happen) those who used TNC1 Rev.3.x digipeaters or Kantronics digipeaters couldn't get their packets thru them. In this area, most of our major digi's are using old VADGC boards with AJ9X code and they

handle Version 2 packets just fine.

Since the BBS runs as an answerer most of the time, the user was in control of which version was in use. Since the BBS forwards mail to other BBS's, and since that forwarding involves digipeaters, it was necessary to put an AX OFF at the start of the FWD.TNC forwarding control file, and restore operation to normal by including an AX ON at the end of the file.

Based on these tests, I have decided to leave the 145.01 input port on W3IWI BBS set with AX ON as a positive step towards future network improvements.

73, Tom

M 5850 Tom Clark (W3IWI,2976) 10/21/85 11:22 PM L:32
KEYS:/A LITTLE FUN/OUR NETWORKS DO WORK!/4 BBS'S LINKED/
TO: (Group 95)

Thought you might find the following message fun -- as a test of Gateway operation tonite I tried a quadruple bounce with the connection described in the text of the message, ultimately reaching N2WX in Florida. I sent a message to Howie and then read it back, and here was the transaction:

\- - - - - - - - - - - - - -

W3IWI de N2WX: Last msg # 2432
(B,D,G,H,I,J,K,L,R,S,T,U,W,X,?) >
r 2432

Msg#	TR	Size	To	From	@ BBS	Date	Title
2432	N	764	N2WX	W3IWI		851021	Heights of futility!

Howie, this probably represents the ultimate packet radio folly.

I am operating TNC2 (yep, with 1.1.1h!) on 145.01 as W3IWI-1 connected to W3IWI. W3IWI BBS is operating as a Gateway to 145.05, connected to K3VPZ
K3VPZ is operating as a Gateway to 14.103, connected to K4NTA
K4NTA is operating as a Gateway to 2 meters, connected to you via STU,MLB
and finally I am logged in on ur BBS sending this message!

Packet radio and our networks do work! And all in all, it is about as fast as EIES is sometimes.

1.1.1h seems to work fine. I like RECON. Haven't explored all the ramifications yet. More feedback later. Good to chat earlier in the day. Good luck on getting the software defined. Posted the SAREX2.004 report tonite on 2978 as an update following a busy day today

73, Tom

W3IWI de N2WX: Last msg # 2432
(B,D,G,H,I,J,K,L,R,S,T,U,W,X,?) >

M 6144 Thomas A. Moulton (W2VY,995) 10/22/85 4:17 PM L:12
KEYS:/AN UNPLEASANT DUTY/BUT WE HAVE A NEED FOR SOME NEW PEOPLE ON FOR A NEW PROJECT IN DRNET/
TO: (Group 95)

I am catching up in my G95 duties...

We have been working on many things,
and one of the projects is going to require a few new accounts,
it is very important and I don't want to have to get out my axe,

I would like to see if there are any users that feel that
their account is not being used to it's full potential and volenteer it.

C2987 CC231 Howard Goldstein (N2WX,2987) 10/27/85 8:58 AM L:36
KEYS:/BLACK HOLE/CANPAC/CTRL Y/CLOCK/DATA LOSS/
A: 230

Tom -

It's probably a pretty good idea to set CANPAC $00 with the MailBox, and it will really disable the action in both CONV and cmd: modes.

I've reproduced another black hole on filling-buffers, but this mode seems okay (i.e. expected). Viz, after eight packets are queued on an individual stream, no more may be queued until something is ACKed.

Internally this is how information-for-packetizing is handled:

1. data pending packetizing? if so, goto (5)
2. process any ^r/^x etc
3. push the character onto the "packet edit" buffer
4. complete packet built in the buffer? No, EXIT
5. insert the packet into this connection's/link's list.
6. did the packet fit in the list from (5)?
   Yes, flag the edit buffer as empty now and EXIT
   No, Set not rdy flag and next time through, go directly from (1) to (5)

Basically what this says is that in CONV mode, editing functions (like ^X) will not work IN CONV MODE while a packet waits in the edit buffer for queuing. Note however that echoing, and CMD: line editing are unaffected by this wait.

Other things: CLOCK- I think I found why the clock was performing so poorly. In fact the clock on my 10.0 RLI is now 2 minutes FAST after being set yesterday afternoon. This poor performance was noted quite early on in the multistream software, I think the cause has been corrected.

CTRL-Y performance: Enhanced ^Y handling so it cancels output on the fly. It impacts on the total throughput of the board since its handled in a real time interrupt and it needs to be tested at high speed before its released like this.

DATA LOSS: Skip's data loss problem seems to be related to - hang on now - echoing, and of all things, the clock. More definitive data later, but it appears that when the echo-processing buffer became full things started slowing down. Enough that asserting hardware flow control was delayed past the overflow limit of the input buffer. 73 Howie

M 8623 Pete Eaton (WB9FLW,2970) 10/30/85 12:28 AM L:29
KEYS:/BACK TO NORMAL NEXT WEEK/
TO: (Group 95)

To: All
From: Pete WB9FLW
Subject: TAPR Phone & TNC 2 Kit Orders
Dist: Open, please give widest dissemination

Chris TAPR's Office Manager is trying her best to ship approximately 700 TNC 2 kits. To add to the problem the label printing program is not working so all addressing and UPS manifests are being done by hand.

To add abit of challange to this "chore" the phone rings...and rings... and rings. Chris cannot do these chore at once or attempt to multiplex them together. So it was decided that the shipment of kits had priority (alot of calls were asking where their kits were). With that decision made it was decided to put the famous TAPR answering system on line to inform folks that called in that Chris was busy shipping kits but would bee happy to take calls after 3:30 P.M. (after the U.P.S. pickup was made). Sounds

simple enough right?......well the answering machine refuses to work, it either does not answer the phone or answers says nothing and hangs up, or leaves the phone off the hook (and give you a busy signal all day long.

The TAPR phone will not be answered until all kits in stock have been shipped and/or it's after 3:30 P.M. Chris will have some extra help in the office next week to help get things back to normal with my arrival on Saturday.

Please let folks know! hopefully things will be back to "normal" by next week.

C2970 CC161 Paul Newland (ad7i,2978) 10/30/85 9:07 AM L:54
KEYS:/EMERGENCY USE OF PACKET/

FROM: paul newland, ad7i
TO: DRAGNET
SUBJ: product review
DIST: open
DATE: 85.10.29

The following posting was taken from USENET. I forward it as a courtesy to the net. The opions expressed here are not necessarily those of only the author.
=======================

Subject: Kantronix in Mexico - Help! + FLAME
Date: 22 Oct 85 02:47:00 GMT
Reply-To: john@anasazi.UUCP (John Moore NJ7E/XE1HDO)

During the Mexico City earthquake disaster, we tried to get a Kantronix packet interface going in Mexico City for health and welfare traffic. Unfortunately, during the short time we had to test it, we couldn't get it to receive anything. The manual was singularly uninformative about troubleshooting. It appears that the FSK work is all done by a magic chip which has no description in the book and a custom part number. We couldn't find any pin that appeared to have the FSK demodulator output.

Does anyone know how to tune this beast up with a scope, rather than just using the frequency counts? Does this thing work? Is it any good? Does anyone at Kantronix know how to write a technical manual?

+++FLAME ON+++
We wasted several hours of valuable disaster net time due to the sorrowful documentation on this product!
+++FLAME OFF+++

What should I tell Carlos (XE1HC)? Should he throw the thing away or is there a rational way to get it to work?
--
John Moore (NJ7E/XE1HDO)
(602) 952-8205 (day or evening)
5302 E. Lafayette Blvd, Phoenix, Az, 85018 (home address)

C2987 CC232 Tom Clark (W3IWI,2976) 10/29/85 11:31 PM L:12
KEYS:/FOR AD7I + WA7GXD/INFO ON TNC2 INNARDS/FOR USE AS DUAL-PORT DIGIPEATER/

Paul, a request, por favor. I am contemplating adapting KE3Z's Dual-port digipeater code for the TNC2 and need some supplementary documentation. Obviously it is ROM low, RAM high, with RAM starting at $8000. Could you provide info on i/o port mapping so I don't have to fish it out of the schematics? Looks like the only change necessary to do the DPD is to swap the SIO/0 for an SIO/2 (to get separate RX/TX clocks on port B), which entails minor surgery on a couple of pins. Comments?

Lyle, I know that the Hitachi CMOS SIO/0 is (sorta) available. Do they also make a SIO/2 in CMOS?

73, Tom
C2987 CC233 Phil R. Karn (ka9q,2979) 10/30/85 2:42 AM L:13
KEYS:/K9NG + TNC-2/

Last week I attempted to connect my TNC-2 to a K9NG modem board. I discovered that the polarity of the DCD line is inverted from what the TNC-2 expects. After fixing that, I discovered that I could generate frames just fine as long as I didn't connect the RXDATA line from the modem back to the TNC-2. If I complete the connection, the TNC will generate one frame and then refuse to send any more.

During transmit the receive data out pin from the modem carries scrambled transmit data. Ordinarily the worst that should happen is that the HDLC receiver sees a few garbage frames. Why should the TNC behave in this way?

Phil

M 9130 Thomas A. Moulton (W2VY,995) 10/31/85 1:18 PM L:12
KEYS:/TNC 2 UPDATE/
TO: (Group 95)

Here is your TNC 2 Update for Today!

Orders up to 1012 have been SHIPPED! (With a couple of exceptions (say 6))

So we should be seeing them show up starting next Wednesday!

They started going out in the mail Last Friday and Chris will be answering the phone today and tomorrow.

If there was a problem with your order she has called you and left word, you can call to check to make sure that yours went out, but she IS calling so don't need to all (500 of us) call her!

M 11126 Harold Price (NK6K,2972) 11/ 5/85 10:02 PM L:65
KEYS:/TNC 2 REV 2/
TO: (Group 95), KV7B

11/05/85
To: ALL, please give widest coverage
Fm: TAPR
Re: Problems with TNC 2 rev 2.

The first of the second shipment of TNC 2s has hit the field, and some problems have been identified.

The following information applies only to TNCs with an ORDER NUMBER greater than 511, those that say Rev 2 below U21.

Symptom: During the test on page 44, the STA light fails to go out.

Solution: The manual fails to tell you to install U24, the second 6264-L. Remove power and install this chip.

Symptom: The digital loopback test on pages 47 & 48 fails. No CONNECTED message is displayed.

Solution: SOME, not all, of the state machine proms (U6) were programmed incorrectly. To see if this is your problem, perform the following steps:

1) Remove power

2) Remove U5(74HC374) and U6(2716 marked STATE) and place on the black protective foam.
3) Install a jumper between U21 pins 13 and 14.
4) Install a jumper between U21 pins 12 and 15.
5) Check for shorts.
6) Apply power, try digital loopback test again.

If the test succeeds, you have a bad U6. To double check, swap U6 with any other working state machine, either from a TNC 2 rev 1 board, a working TNC 2 rev 2, or the K9NG 9600 baud modem board.

If your U6 is bad, put it on a small piece of the black foam, put the chip and foam in the small box your sockets came in, and send the box back to TAPR. We will return it asap by first class mail.

Symptom: Your front panel is missing.
Solution: No one got one. A sheet notifying you of this was left out of the box. The panels will be sent to you via first class mail by the end of the week.

Symptom: Your board says Rev 2, but the manual says Rev 3.
Solution: Replace Rev 3 with Rev 2 in the manual. There is no Rev 3.

We apologize for these problems. Remember, the preceding applies only to TNC 2 order numbers greater than 511 (rev 2). Owners of TNC 2 rev 1 (order numbers less than 512) will want to be sure to see the next PSR, it contains information on how to upgrade your TNC to rev 2. PSR should be mailed in the next 7 days.

TAPR

M 11141 Tom Clark (W3IWI,2976) 11/ 5/85 10:56 PM L:21
KEYS:/TNC2 REV.2/ERRATA/TESTIMONIAL/
A: 11126 TO: (Group 95)

In Harold's note on TNC2 Rev.2 he referred to U6 as a 2716 EPROM. I believe that U6, the state EPROM, is being shipped as a 27C64 in most (if not all) of the Rev.2 TNC2 kits.

As a small testimonial, I played "beta" site for both the Rev.1 and Rev.2 boards. The subtle changes in Rev.2 really do improve the operation. The all-CMOS design plus the mounting of the 7805 eliminate a heat problem experienced with the Rev.1 board. Grounds and power supply traces and bypasses have been improved to improve noise immunity and cut down on RFI. A fifth LED for PTT has been added. And both the ROM and RAM have been doubled; this in turn let our software guru, N2WX exercise several improvements over the earlier software -- things like multiple connections, digipeater call aliases, the ability to re-establish a path thru different routes without breaking the link, etc. are all super!

Those of you with Rev.1 boards will definitely want to make the mods that Harold referred to -- they involve primarily changing the 2 2764 EPROMS for one 27C256, and then changing the (now unused) 2nd EPROM socket to take a second 6264 8k RAM chip -- resulting in 32k ROM + 16k RAM in 3 chips.

73, Tom
C2973 CC38 Phil R. Karn (ka9q,2979) 11/ 7/85 2:28 AM L:43
KEYS:/TCP WORKING!/

I am happy to announce that my TCP code is functional. I successfully

carried out its first real workout tonight in a "QSO" with my vax at work. The path looked like this:

IBM PC/XT -- my code, originating end of TCP connection
1200 baud asynch port (using SLIP for datagram encapsulation)
Ven-tel 212 modem
dialup phone line
Penril modem (company modem pool)
Micom data switch
T-1 interlocation link
Micom data switch
1200 baud asynch hardwired line
asynch interface on my office SUN with SLIP driver
SUN Workstation running 4.2BSD; acts as IP gateway
3-Com Ethernet interface
Local area network
3-Com Ethernet interface
Inter-Ethernet IP gateway (another SUN)
3-Com Ethernet interface
Another local area network
3-Com Ethernet interface
VAX-11/780 running 4.2BSD --- other end of TCP connection

The whole setup worked remarkably well. Of course, the slowest part of the path was the dialup phone line. I noticed that the line was fairly noisy. Whenever a SLIP-encapsulated IP datagram was sent over it after the line had been idle for a while, I noticed that it would often not make it through. This is because SLIP does not put any framing characters on the beginning of a datagram, only at the end, and should a noise character occur between datagrams, the datagram that follows will have it tacked on the beginning. Of course, the checksums at the IP level detect this and discard the packet, which TCP later retransmits.

For demonstration purposes, it would be trivial to replace the modem link with a pair of TNCs so that SLIP-encapsulated IP datagrams could be sent over the air without worrying about the AX.25 I-frame boundaries. Of course, since Dave Mills has already beat me to the first TCP/IP contact over AX.25, I don't see much point in duplicating that demo...

Phil

M 11822 Jack Brindle (WA4FIB,2963) 11/ 7/85   9:44 PM L:30
KEYS:/IT'S HERE AND IT WORKS!!!/
TO: (Group 95)

To: all who care.

I GOT MY TNC2! "They" are right. it is a neat machine. After having UPS (or someone) misplace one, I got mine shipped out pronto! It took me a day to assemble, then half the next to fix the bad state machine rom problem. For those with that problem, the intel hex code for the state machine is located in the linking conference as CC26 or CC27. After programming a 2764 with that code, the thing started right up and actually received data! The 27C64 was then erased and reprogrammed with the correct info and it works! I ran some current measurements and found the draw to be only about 70 milliamps! It would be interesting for others to measure the current draw on their TNC2s. One problem I ran into was that I didn't clip the leads on top of the on-off switch, and sure enough it shorted to the case. After clipping them, everything is fine.

For those of you trying to interface your TNC2 to a Macintosh, be sure to ground pin 1 on the TNC2 DB25 connector. Pin 1 is supposed to be "chassis ground". Unfortunately it is not connected to anything! A lot of cables use this pin for ground instead of pin 7, so it must be tied to ground! The reason for the problems in using the TNC2 with the Mac is that many

(Your comments ...
Mac - RS232 cables use pin 7 of the DB25 for RX data ground, and pin 1 for signal ground. Most, if not all, 103/212 modems have pin 1 grounded to allow this configuration. Also note that the chassis is grounded only through the ground tab on the +5 volt regulator. Anyway, after making this ground connection, everything runs fine! Now I just need to learn all those new commands and parameters. Nice job, Howie!

On another note, I have now moved from Florida to the Atlanta, GA area. my address here is: Jack Brindle 2980 Wayward Drive Marietta, GA 30066. Phone: (404) 565-6002.

More later. I want to "play" with my new TNC2!

- Jack B.

M 11825 Jack Brindle (WA4FIB,2963) 11/ 7/85 9:51 PM L:9
TO: (Group 95)

One more thing - Thanks a bunch to Pete for his help in "finding" my lost TNC2 and explaining the state machine difficulty. Perhaps we should double his salary instead of cutting it...
(Smile Pete. Remember, "Wait'll next year...").

A word of caution, though. One can get addicted to watching the STA LED after you send a packet to see when the ack is received. Also, notice how quiet the TNC2 is - I had to place the antenna of my IC2AT right next to the PC board to get any noise! Nice job, guys!!!

- Jack B.

M 12380 Pete Eaton (WB9FLW,2970) 11/ 9/85 9:44 PM L:21
KEYS:/IMPORTANT INFORMATION FOR ALL TNC 2 OWNERS!/
TO: (Group 95), WB9FLW

To: All
From: TAPR
Subject: Important TNC 2 Information!!
Dist: Open

Feedback from the field has shown two serious problems that ALL TNC 2 owners should be aware of!

If the Lithium battery is not mounted flush against the board and/or the PC board is warped it is possible that the case can be shorted to ground. All owners should insulate the top of the battery with a piece of electrical tape.

Notice to owners of Radio Shack Model 3 and 4 remove R 14, 15, & 16 these resistors have been shown to draw excessive current to the point of damaging the board! Please open your unit and make this modification immediately.

M 12482 Gwyn Reedy (W1BEL,2975) 11/10/85 7:58 AM L:21
(ORIG.) 11/10/85 7:35 AM L:15
KEYS:/PACKET EQUIPMENT ANNOUNCEMENT/
TO: (Group 95)

TO: The Packet Radio Community (Wide Distribution Encouraged)
FROM: Gwyn Reedy, W1BEL
SUBJECT: New Packet Radio Enterprise

.

Andy DeMartini (KC2FF) and I have started an enterprise to support the packet radio community. We will be doing business as Packet Radio Systems in Tampa, Florida. This is a private venture, fully separate from our volunteer work with TAPR and FADCA. Packet Radio Systems is purchasing the OEM kit for the TAPR TNC2 and will very shortly have available TNC2 bare boards, hard-to-find parts kits, full kits, cabinets, etc. Prices will be set with the builder and experimenter in mind. Other products are planned for the immediate future. Watch for our advertisements for details on these items to support the innovators in Amateur Radio.

.

(Your comments are welcome. I intend to continue working for FADCA and TAPR in a volunteer capacity. This new venture won't have any effect on my editorial policy with the publications I edit! Gwyn)

M 12720 Gwyn Reedy (W1BEL,2975) 11/11/85 6:16 AM L:22
KEYS:/PACKET RADIO MAGAZINE/
TO: (Group 95)

ANNOUNCING PACKET RADIO MAGAZINE

.

Beginning in January, the FADCA > BEACON will become Packet Radio Magazine. The publication will feature: o Larger, Clearer type o All pages stapled, booklet style o A two color cover on heavy stock o More pages, more news, more ads o Sections devoted to news from other areas of SOUTHNET and other parts of the country.

.

The purpose of the change is to allow this publication to better serve those readers and potential readers outside of Florida. Organizations smaller than FADCA will be able to have a large monthly publication for their members that includes a locally produced section. Good technical and operational news will be uniformly available to a broader group of packeteers. In effect FADCA will publish a 'group newsletter' for all those organizations that desire to participate, and all concerned will be able to share in the reduced costs of quantity printing and a broader selection of authors. If you think your organization could benefit from participating in this venture, contact FADCA at 812 Childers Loop, Brandon, FL 33511 for further details.

M 13042 Lyle Johnson (WA7GXD,2973) 11/12/85 12:07 AM L:17
KEYS:/OFFICE HELP REVISITED/
A: 11506 TO: (Group 95)

Rich,

.

Regarding the problems you mentioned with the office staff in M 11506.

.

I have given TAPR's EVP a stern lecture on TNC 2 Rev 2's HS-CLIP-1 (an innovation slightly inferior to the ALJ-1000, but still very useful...) and its proper application in Amateur packet radio.

.

I further have docked him two-weeks pay and withdrawn his all-expenses paid trip to the 1st Bermuda Conference on Packet Radio to be held the entire month of January on a lovely island...

.

Thank you for bringing this matter to my attention. It is only through such public 5th column activities that we can truly know the nature of our officials.

.

Lyle

M 13062 Dan Morrison (KV7B,2986) 11/12/85 1:01 AM L:26
KEYS:/CORRECTION TO PSR REV 1 MODIFICATIONS/
TO: (Group 95)

Having found my name cited as one of two authors of modifications to TNC-2 rev 1, I thought, what the heck, I'd try them out. Well, I'm here to report that there's been a slight misprint in the PSR article on upgrading rev 1 to near rev 2 condition. Unfortunately, it's an extremely important error, causing the revised board not to work after revision. Fortunately, it's an easy error to correct if you have been successfully cutting traces already. Here's the correction:

Revision F, Memory modification, leaves out two steps in the conversion of socket U24 to accept a 6264. After steps 6 and 8, add the following two steps:

6a On the top of the board cut the A13 address line to pin 26 of U24.

As examination of Fig. 1 of page 10 of the October PSR shows, this trace is easily located as the second trace from the top, within the U24 outline. Be careful not to bugger up any other traces as they are kinda close.
8a In step 8, continue the jumper to pin 28 over to pin 26 as well (i.e., have pins 28 and 26 shorted together by the jumper you install leading back to U25).

These two steps should be performed in the order indicated, as the second step will cause high current drain on the 3 volt battery unless the A13 line has been cut first (ask how I know this...).

Good luck and happy slicing, you rev 1'ers.
73
Dan

M 13100 Gwyn Reedy (W1BEL,2975) 11/12/85 6:35 AM L:11
(ORIG.) 11/12/85 6:27 AM L:10
KEYS:/TNC2 STYLE KITS/
A: 12482 TO: (Group 95), W1BEL

Message 12482, Packet Equipment Announcement, has been modified. The company name is now PacComm Packet Radio Systems, Inc per advice of our lawyer. Also we have extablished some phone numbers. Here they are for your information (and distribution, if you want): Tech info - (813) 689-3523

Orders: 1-800-835-2246 ext 115 (In Kansas 1-800-362-2421 ext 115). The order numbers are exactly that, a telephone clerk with a pad of paper. She won't know a TNC from a snowball, nor anything about availability etc. We are looking to have the shipments of TNC2 'look alikes' started in December. See the ad in the November BEACON for part numbers, or write to PacComm Packet Radio Systems, 4040 W. Kennedy, Tampa, FL 33609. 73, Gwyn

C2973 CC40 Phil R. Karn (ka9q,2979) 11/13/85 1:07 AM L:15
KEYS:/TCP/BOLOGNA IS A CITY IN ITALY/

Jack, I returned from the AMSAT meeting in Colorado last night, so I haven't been able to read and respond to EIES until now.

As you should know by now, the TCP/IP protocols are highly modular and can be installed very easily on top of many different link level protocols. I got it running first with SLIP so that I could test it out with a "real" machine at the other end, and the next step is to integrate it with the AX.25 Level 2 code I've already written. The interfaces below IP are very easy to change, because of the simplicity of the datagram interface.

If you ever send me a SASE, I'd be happy to send documents and a copy of both my AX.25 code and my new TCP code.

Phil

C2974 CC101 Lyle Johnson (WA7GXD,2973) 11/12/85 11:56 PM L:189
KEYS:/NNC UPDATE/GET OUT YOUR ASSEMBLERS/

Here is some preliminary information on the TAPR NNC digital hardware

.
MEMORY
.
There are 16 bytewide sockets. The initial units will be expecting a 27C256 EPROM in the lowest-order socket. That socket, and the next three, will be strapped for 32k-byte EPROMs and are mapped to the bottom 128k bytes of memory (00000H - 1FFFFH).
.
The next four sockets, 20000H - 3FFFFH, are strapped for 32k-byte RAM ICs. By careful use of address mapping in the software, 8k-byte RAMs may be substituted.

.
The above eight sockets are designed to have a single wait-state inserted during opcode fetches to allow use of 300 nSec or slower memory with the normal clock.
.
The next eight sockets are designed to accept 8k-byte or 32k-byte RAM ICs. The default strapping is for 8k-byte parts. This memory area starts at address 40000H and extends to 4FFFFH (8k-byte ICs) or 7FFFFH (32k-byte parts). These sockets have no wait states, so plan on using 150 nSec or faster RAMs if code is to be executed from this area. These sockets are battery-backed.
.
Note that if 8k-byte parts are used, any memory test/sizing program must take into account the fact that the memory will "repeat" every 64k bytes!
.
CLOCK
.
The battery-backed clock was based on the MOSTEK 3835. UTC just dumped MOSTEK, so there will be no clock on the first boards. Anyone know of another 8-pin CMOS clock?
.
ASYNC I/O
.
There are two serial ports on the 64180 procesor.
.
The CON: async port has TxD, RxD and CTS (to the 64180) as well as SG signals only.
.
The AUX: async port has TxD, RxD, CTS, RTS, DCD (CTS and DCD being inputs) and SG signals only.
.
With a standard clock of 4.608 MHz (using a 9.216 MHz crystal), the supported baud rates on these ports will become 150, 300, 600, 1200, 2400, 4800 and 9600.
.
I/O MAP
.
I/O off-chip to the 64180 is mapped as follows (addresses in hex):
.

80-9F:	FDC !SEL (uses 9266 Floppy Controller as option, same commands as NEC 765A)
A0-BF:	FDC !ACK (DMA ACK) - to ack a DMA request to the FDC
C0-DF:	PIO !SEL
E0-E8:	SIO #1 !SEL
E9-EF:	SIO #2 !SEL
F0-F8:	5380 !SEL
F9-FF:	5380 !ACK

.
Please use the lowest order addresses for all software mapping. For example, the FDC has only a single address line, so it occupies only two bytes of the 32 bytes allocated for it. Use addresses 80 and 81 (hex) for accessing this part; this will allow for future expansion in an orderly manner.
.
The FDC is mapped to DMA channel 1 when used, and the !INT2 interrupt input on the 64180 is reserved for the FDC. !TEND1 is inverted and applied to the 9266 as TC. A clock divider is used to provide the required delay between the 9266 asserting a DMA request and the hardware passing the request on to the 64180.
.
The PIO is a Z80 PIO, not an 8255. This allows the use of mode 2 interrupts. The "A" and "B" ports are tied to 26-pin headers to allow an IDC cable to be connected pin-for-pin to a Centronics compatible printer interface.
.
The SIOs have the C/!D line tied to A0 and the B/!A line tied to A1.

SIO #1 has the highest priority in the daisy-chained I/O, followed by SIO #2, then the PIO.

.

by means of push-on shunts. Similarly, !RDYB of SIO #1 may be jumpered to 64180 DMA channel 0. !SYNC lines of the SIOs are unterminated. SIO

.

All four SIO ports are brought out to 16-pin headers. All even pins are grounded. Odd pins are mapped as follows:

.

1 - TxD
3 - TxC
5 - !RTS
7 - !CTS
9 - !DTR
11 - !DCD
13 - RxC
15 - RxD

.

The pin descriptors are those of the SIO pins, not any RS232 standard. These connections are unbuffered.

.

SCSI

.

The NCR 5380 SCSI chip is used. !INT1 is the channel used for SCSI. DMA Channel 0 is defaulted for DMA operation, but SIO #1 may optionally be selected for this DMA channel as mentioned above.

.

is inverted before application to the 64180.

.

MISCELLANEOUS

.

The floppy disk adapter uses a 34-pin header for pin-to-pin wiring to 5.25 and 3.5 inch floppy drives. No provision is made for 8 inch drives. (It is 1985...)

.

The CON: and AUX: ports use a 20-pin header to allow a crimp-on DB25 for RS232 use. CON: looks for a DTE, AUX: a DCE.

.

The whole NNC digital controller and floppy adapter will run from +5 volts. All boards are four-layer. The digital controller is smaller than a TNC 2 !! (44.6 sq inches vs. 48.6 sq inches)

.

The system is designed to boot and run the Z-system port done for the SB180.

.

AVAILABILITY and PRICE

.

I am working on prototype number 1 right now. Assuming that it works (uP accesses memeory and I/O, etc.), it will take 5 weeks to get the first lot cranked out. I am inclined towards a wave-soldered board, so that will add 1 to 2 more weeks. If it takes two weeks to be sure everything is working, that makes about 9 weeks for Beta test versions of the digital and floppy boards. The Modem boards will be about 2-3 weeks later.

.

Tentative -- very tentative subject ot change etc. etc. -- pricing looks like the following:

.

NNC digital board w/32k EPROM, 64k RAM, no SCSI chip, no clock/calendar $175.

.

Floppy adapter $125 (includes $50 Z-system and docs).

.

Modems (4 modems, 1 tuning indicator, etc.) $150.

.

I am hoping to revise this downward, but the three PC boards alone in 100 lots will cost us $100 (four layers don't come cheap!).

.

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

I would like to suggest we standardize on 5.25" floppies rather than 3.5". Question: should we go for 96 tpi (770k per drive) or 48 tpi (386k) drives? I lean towards 96 tpi...

.

TASKS

.

1) We need an AX25L2V2 "kernal" optimized for SPEED. An interface spec needs to be written. I like the idea of Phil's suggestion -- pass it a pointer and a length or ??? and let the routine handle ALL the level two stuff. Then interrogate amemory locaion (pseudo-register) to see if it was passed OK (or maybe increment/decrement a semaphore). This should be an interrupt-driven, DMA-able routine, callable by several callers (re-entrant, maybe recursive?). This one needs a lot of careful crafting.

.

2) A multi-tasking kernal would be very useful.

.

3) Since we have four HDLC ports, and each should be able to handle multiple logical connections, the ax25 handler mentioned above needs to be very carefully designed so it won't get confused.

.

4) All code generated should be ROMmable. The FDC adapter is an OPTION for developers, not a requirement for operation.

.

5) I am going to propose to the TAPR Board of Directors that the design and rights to the NNC be placed in the public domain, subject to the usual disclaimers of liability, etc. I would like to request that anyone doing software for this project make the source code available on the same basis.

.

CLOSING

.

I need a count of folks who are wanting to participate in the further development of this project. This isn't Beta-testing a system; this means developing the system that will eventually be beta-tested! It will mean purchasing an NNC, probably at prices near those guestimated above. It will mean lots of hours when you'd rather be doing something else. Lots of software development time. Debugging. Testing. Checking other code from others working on the project. Possibly having all your work go for nought.

.

And making a contribution to a project you'll be proud to tell your grand-children about...

.

Onward!

C2974 CC102 Phil R. Karn (ka9q,2979) 11/13/85 2:21 AM L:33
KEYS:/NNC SOFTWARE/

Lyle, that's good progress on the NNC!

A few comments on your points.

My boss at Bellcore wrote a nice small "kernel" in C for some multitasking applications. It is modeled after Intel's RMX-80 but is much more portable; it has been used on PDP-11's and 68000s for quite a few standalone projects in our group. Since he wrote it while working on a PhD, he "owns" it and he's given permission for me to give it to this group. The only work involved in "porting" it to the Z-80 is the writing of a few assembler primitives like a task switch routine. All processes live in a common address space and synchronize with message passing through structures called "exchanges". This kernel is about 2K on a PDP-11, so it shouldn't get too bad on a Z-80.

I believe my 820 TNC code should be easily adaptable to the NNC. I wrote it with minimum CPU loading as the prime criterion, in that

I spent a lot of time designing the data structures and algorithms to avoid memory-to-memory copies whenever possible. Packets consist of linked lists of "message buffers", which are threaded together as each protocol layer adds its header to a packet. One memory-to-memory copy (or its equivalent) is still needed when the packet finally goes to the device, unless the DMA controller is smart enough to do scatter-gather I/O. Does the NNC DMAC do this?

I also believe my code should work well with multiple connections across multiple link controllers, since I designed it with that in mind and have it running on an 820 with two HDLC channels on a FADCA board.

Phil

M 15973 Wally Linstruth (WA6JPR,2989) 11/20/85 1:45 AM L:82
KEYS:/IP TEST BED/820 QUESTION/
TO: ka9q, KA6M, WB6HHV

Hi Phil, Hank, Mike, Phil, I recently contacted Mike and Hank regarding my desire to help with the protocol wars. I had hoped we could start at least an outline of a high level spec for incorporating IP/TCP as the next iteration for protocol experimentation. It seemed that the nature of the current wars was kind of hit or miss with everyone defending his current "hot button" but with no one taking a broad but detailed look at over- all needs vs. capabilities. I had hoped we might have at least an overall outline ready for the Newington meeting this December. I then talked to Paul Rinaldo about the same subject but he indicated that there would be no way to sell using DOD protocols to the League because the Europeans simply wouldn't buy the deal.

In the mean time, I had talked with Hank and Mike about both the spec and some experimentation. Hank is quite busy and felt he could not be much help with spec writing but he did kind of like the idea of doing some IP stuff using UI framing. He suggested that that might be a good use for Vancouver boards if software was rewritten to just perform as UI sender/receiver. The Vancouver board to act as a packetizer (framer) for IP frames coming out of a machine such as an IBM PC. He also felt that he might be able to help convince some folks in Westnet (Site control folks) to support some field testing.

I also recently became acquainted with a couple of live wire types here in Santa Barbara who would like to support some testing and I know that Mike in San Diego would also like to help. Additionally, Mike was going to hit on Brian Kantor - WB6CYT to see if he could lend a hand (He is just finishing his degree and may be too busy but we'll see).

We here in S. B. have come up with some surplus 900 MHz gear which may also be useful.

I would very much like to try to set up a testing mechanism for IP - TCP - TELNET etc. here in California. I think we may be able to really do some good.

There is some new hardware out in the commercial world which may be useful also. I have just ordered a 10 MHz 68000 co-processor for my PC from HSC. It has no comm chip for bit oriented protocols but the 68k bus is available on a header for the addition of off board goodies. I got a good price for the card with os9/68 operating system (unix like) based on my desire to use the card for feasibility studies for networking. The whole thing was $500.00. I also am scheduled to attend a one week training session (starts Dec 2) at IBM Boca Raton for an as yet unannounced communications co-processor for the PC. This one has a 186 processor, 2 - 8030s, a parallel chip and 256k ram with an honest multi-tasking operating system with hooks into PC network, Net-bios, etc, etc. I should have lots of DATA AT NEWINGTON.

FURTHER.....BASED ON THE EXCITING RESULTS REPORTED BY W3HCF, I HAVE BEGUN THE PROCESS OF RETIRING MY TAPR BETA BOARD IN FAVOR OF A TNC1 WITH DED FIRMWARE. I

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

HAVE ALSO BEGUN TO SERIOUSLY WORK ON BRINGING UP ONE OF MY 820 BOARDS WITH THE PHIL SOFTWARE I PICKED UP LAST MARCH. I AM HAVING DIFFICULTY CONNECTING TO MYSELF VIA CHANNEL B WHICH IS LOOPED BACK ON ITSELF...NO MODEM OR RADIO. SENDS FINE BUT NO RECEIVE. I HAVEN'T HAD A CHANCE TO GET AFTER THE PROBLEM YET TONIGHT BUT I THOUGHT I'D CHECK WITH YOU FIRST.

ANYHOW........I GUESS I'M OFFERING TO PRESS ON HERE TO GET A GEOGRAPHICALLY SEPARATED TEST BED UP FOR IP EXPERIMENTS AND NEED TO KNOW A FEW MORE THINGS. HOW MUCH POSTAGE FOR YOUR LATEST STUFF PHIL. BETTER YET, I'LL SEE YOU IN NEWINGTON WITH DISKS TO TRADE?? I WOULD LIKE ALSO TO HEAR FROM MIKE AND HANK ON THESE IDEAS AS IT'S BEEN A FEW WEEKS SINCE WE TALKED.

CAN WE DO SOMETHING COLLECTIVELY? I WOULD LOVE TO GET SOME REAL WORLD FEEDBACK ON THE EFFECTIVENESS OF IP IN THE AMATEUR RADIO ENVIRONMENT IN SPITE OF THE LEAGUE'S POSITION.

73 FOR NOW WALLY

M 14943 JEFF WARD (ARRL,2977) 11/17/85 12:36 AM L:10
KEYS:/GREETINGS FROM NEW GATEWAY EDITOR/
TO: (Group 95)

Hello All:

Just wanted to let everyone know that I will be using Jeff's account for the next few days to gather information for the next issue of Gateway.

Thanks,

Ed Raso, WA2FTC

M 15192 Jack Brindle (WA4FIB,2963) 11/17/85 8:31 PM L:30
KEYS:/SESSION LAYER PROTOCOL!/
TO: (Group 95)

For those of you who have been impatiently awaiting my session protocol specification, your wait is about over. I have basically completed work on the initial protocol spec, & will indeed "unveil" it at the Southnet II packetfest next weekend. The spec is a bit long, about 15 pages. Part of this is due to being done with MacWrite, but mainly it is because of the implimentation discussion it contains. So what will the session layer do? Quite simply, it will allow you to establish "sessions" (otherwise known as connections) with one or more stations. In other words, it allows you to converse with someone while transferring files with him. Depending on the implimentation, you may not even see a delay in the QSO due to the simultaneous file transfer! (Note to layer 3 implimentors - this is the real reason priority is so important in what you are doing!). The multi connects allowed by several TNCs also works into the session layer's capabilities, since it now gives a good means of controlling and seperating those connections. This protocol also provides an excellent base for file transfers - just add an overlying protocol to send file name and parameters (for MacPacket, this is the MacBinary protocol). Additionally, the session protocol supports transfers for all kind of devices (now you will know just what you are talking to), and is extensible for future developments!

My presentation on the proposed session layer is scheduled for 9AM next Saturday (the 23rd I believe) at the Southnet II packetfest to be held on the campus of Georgia Tech here in Atlanta.

Note to sysop - the spec is 15 pages long (might be shorter as unformatted text only, maybe by a page or so). Is this too long to upload (yes, I know about big telecom bills)? If not, where would I put it? In a conference (we seem to have taken over 2973 with this stuff)?

Anyway, if you are interested, but cannot attend, let me know, I will try to get a copy of the spec to you - let's see if we can get it up here first, though.

73, Jack B.

M 15239 Tom Clark (W3IWI,2976) 11/18/85   1:01 AM L:22
KEYS:/W3IWI PTT MOD FOR TNC2/CORRECTION/EATING CROW/
TO: (Group 95)

My modification to the Rev.1 TNC2's has now appeared in several places, including PSR and CTM. This note is to offer a caution, caveat emptor, and to eat a little crow!

In the last paragraph I indicate that you can reverse the sense of the LED from bright=rcv, dim=xmit by attaching the 1k resistor to pin 9 of U14 instead of pin 8. >>>>>> DON'T DO IT!!!!! <<<<<<<

The problem stems from the fact that U14 is used as a part of the BBRAM logic and is powered from the battery when the power is off. If you reverse the sense, then you will be driving battery current thru the LED and will kill your battery in short order.

The mod as I wrote it up (dim on xmit) works fine and doesn't hurt the battery since when the power is off the U14 pin 9 output is low and no current flows thru the LED.

Hence, please delete the last sentence of the last paragraph of my modification notes and forget I ever said it!

73, Tom

M 15242 Tom Clark (W3IWI,2976) 11/18/85   1:05 AM L:51
KEYS:/COMPUTER TRADER MAG/CTM/PACKET SWEEPSTAKES CONTEST/PRIZES WITHDRAWN/NOW PLAQUES/
TO: (Group 95)

Following all the negative comments on DRNET (which were reflected in W1BEL's editorial in FADCA> BEACON) about Computer Trader Magazine (CTM) offering merchandise prizes with their Packet Sweepstakes contests, the following appeared as a loose sheet folded into my October copy of CTM:

"   IMPORTANT ANNOUNCEMENT --- PLEASE READ"

" When we originally came up with the concept of the CTM Packet Sweepstakes, we were as pleased with the concept as many of you were. Many hours of discussion, planning and thought were put into its design. This work included many hours of discussion of the possible impact such a contestcould have on packet radio as well as our whole amateur society. This possible impact could best be summed up by Gwen (sic) Reedy, W1BEL, in his column in the October issue of the FADCA> BEACON."

" In this article Gwen says 'Leadership and initiative in the packet movement are gradually passing away from the experimenters and organizers and into the hands of the businessmen. As that happens, the special charm that separated packet from the more established ham radio pursuits begins to fade. There is nothing wrong in what is going on, but it is a bit sad. Rather like the passing of the early days of the microcomputer, the charm and excitement are not there any longer; they have been displaced by the sounds of coins'."

" This was not our intention when we conceived the CTM Packet Sweepstakes. We honestly were trying to give packet radio a big shot in the arm. We were totally unaware that we might be giving the patient an overdose. There was no profit or money involved in the contest. Every prize was to be donated and CTM was only to serve as a gathering place for contest participants."

" Those of us at CTM do not want to commercialize packet or ham radio. All of us, just as so many of you, have spent many hours promoting its growth and educating others to its beauty. The charm that Gwen wrote about should be protected at all costs. It is the one ingredient that will keep our hobby

alive. It has been many a year since something like packet radio has come along and provided such a positive impact - such new 'life'. Therefore, it is that we must withdraw all the prizes that we listed. The contest, however, will still go on. In place of the prizes, we instead will offer beautiful engraved plaques for the first three places in each of the three contests. All who achieve these goals will receive beautiful certificates."

" You don't know how much it hurts to do this to you. We wanted to give these things just as badly as you probably wanted to win them. However, you must remember that all of us at CTM are hams. We believe in the sanctity of ham radio and all that it stands for. It is this spirit that we at CTM want to promote and are striving to provide the best possible magazine for packet radio as well as other ham radio and computer activities."

" /s/ CTM 1704 Sam Drive Birmingham AL 35235 (205)854-0271 "

M 15701 Paul Newland (ad7i,2978) 11/19/85 12:20 PM L:20
TO: (Group 95)

FROM: paul newland, ad7i
TO: all TNC2 Rev 1 and Rev 2 owners
SUBJ: TNC2 latch-up
DIST: open -- please distribute far and wide
DATE: 851119

We have found a problem that appears on a very small number of TNC2 boards. Sometimes, the MF10 switched capacitor filter will latch-up to the positive +5 supply causing the regulator to get very hot. A solution to the problem is to change C10 on the -5V supply from 10uF to 47uF. This cures the problem on all the boards we have seen. If you experience any latch-up after making C10 47uF, please let us know immediately. Given our results, I strongly recommend that all TNC2 owners, whether they have the latch-up problem or not, change C10 from 10uF to 47uF, regardless of REV number. Thanks!

73, paul

C2973 CC41 Phil R. Karn (ka9q,2979) 11/16/85 1:50 PM L:13
KEYS:/A WORD FROM THE EXPERTS/

I had a chance to chat with Professor Dave Farber of the Univ of Delaware yesterday when he visited Bellcore. He was one of the members of the NRC panel that issued the TP-4/TCP report.

I told him about the amateur packet radio protocol wars, and he commented that he certainly wouldn't hold up our progress by holding out for TP-4. The ISO protocols are so slow in coming that he has no problems recommending TCP for systems being developed and deployed now. The likely time scale for TP-4 deployment is at least 5 or 10 years away, although sometimes I wonder if the sun won't be in its red giant stage before it actually happens...

Phil

M 16146 Phil R. Karn (ka9q,2979) 11/20/85   2:03 PM L:29
KEYS:/PROTOCOL WARS/
TO: KA6M, WA6JPR, WB6HHV, ka9q

Wally, what's this about Paul saying that the Europeans would never "buy" TCP/IP. How come? Is it because it isn't a CCITT protocol? Neither is AX.25, despite the name, but there doesn't seem to be a problem with it in Europe. Is it because the DoD uses it? Tell him that I'm a pacifist too, but that doesn't prevent me from pulling some civilian "salvage value" out of the trillions we spend on defense. And just who are the "Europeans" who say this, anyway? I wonder if Paul has been listening to Gordon Beattie a bit too much, since this is one of his favorite themes. The effect of "disqualifying" TCP/IP would be to eliminate the only practical, proven and effective datagram protocol from consideration; the TP-4 "alternative" is still just a (changing) paper tiger, and the sun is likely to be in its red giant stage before that thing becomes anything else.
I'm pretty upset about this, because all along I have worked on the assumption that the best protocol from a technical standpoint would have the best chances of being adopted, and now you tell me that political considerations make my efforts futile. ARGH!!!

In any event, I have spent a lot of time recently polishing off a TCP implementation written in C. As I've said in 2973, I've successfully tested it over an asynchronous serial line into the ARPA Internet, establishing connections and logging in to machines. Read also my comments about SLIP; this is a way that TCP-based hosts and IP/AX.25 packet switches could be separated into different boxes without requiring HDLC interfaces on everything. All you need to run my TCP is a machine with a conventional asynch RS-232 port, and that line connects to an 820 which does packet switching and also has the AX.25 channels.

Phil

M 16287 Wally Linstruth (WA6JPR,2989) 11/20/85 11:43 PM L:52
KEYS:/POLITICS/820 QUESTIONS/
TO: ka9q, WB6HHV, KA6M

HI PHIL,

I'M MYSTIFIED TRYING TO SEE THE LOGIC OF PAUL'S STATEMENTS REGARDING TCP/IP RE THE "EUROPEANS". I HAD ACTUALLY CALLED HIM TO DISCUSS TRAVEL ARRANGEMENTS FOR THE MEETING AND TO SEE IF HE WAS INTERESTED IN HAVING AN IP SPEC FOR THE AGENDA. HIS RESPONSE TOOK ME BY SURPRISE AND I DIDN'T PURSUE IT IN DETAIL AT THE TIME. HE SAID IT SO MATTER OF FACTLY THAT I REALLY WONDER IF HE COMPREHENDS THE GRAVITY OF WHAT HE SAID. I TRUST YOU WILL BE AT THE MEETING.

I DON'T THINK THAT POLITICS WILL BE THE SOLE ARBITER OF THE ULTIMATE PROTOCOL DECISION, AT LEAST IT HAD BETTER NOT BE. I WILL CALL PAUL IN THE NEXT DAY OR SO TO GIVE HIM MY TRAVEL ARRANGEMENTS AND WILL TRY TO GET SOME FEELING OF WHERE HE'S REALLY COMING FROM.

PROBABLY BETTER TO SEE IF IT WAS A REAGAN STYLE "GOOF" BEFORE WE START POUNDING ON THE TABLE.

73 - WALLY

M 16266 Jack Brindle (WA4FIB,2963) 11/20/85 10:20 PM L:20
KEYS:/SESSION LAYER PROTOCOL/
TO: (Group 95)

Note to all about my session layer specification.
I will be uploading the specs as soon as I figure out how to work with

and support organization ...
... be uploading the specs as soon as I figure out how to work with
notebooks. Apparently this is the best way to post the spec, so it will be the way I use. I will hopefully be uploading it right after the weekend, when I get a chance to put it into a more readable form (electronically at least). The document is in Macintosh MacWrite format, and would be almost impossible to read on a "normal" terminal.

I am releasing the spec at this time to help in file transfers and the coordination of multi-connect qsos (both with a single station and several). Although it may duplicate some of the functions of TCP/IP, it is needed now, and there is no reason to delay until a network scheme has been chosen. The session layer protocol runs with both networked AND point-to-point systems, so it indeed can be used immediately. In fact, my MacPacket code will soon be using it for binary file transfers. The only assumption that the protocol makes is that a network and transport layer will someday be used, so that "placeholders" are placed between the link layer header and the session header. These placeholders consist of a pair of NULLS (hex ØØ), indicating a GFI of zero (network layer) and a TPDU length of Ø (transport length).

Watch for the upload, probably to occur early next week.
73, Jack B, WA4FIB.

M 16663 Tom Clark (W3IWI,2976) 11/22/85 2:16 AM L:29
KEYS:/WØRLI CLONE FOR IBM-PC/TURBO PASCAL/DE WA7MBL/
TO: (Group 95)

The following is forwarded from the W3IWI and W3VH BBS's:

```
W3IWI at 851122/Ø433: Last msg # 959Ø, 73 active msgs
(B,D,G,H,I,J,K,L,R,S,T,U,W,X) >
Msg# TR  Size  To     From    @ BBS   Date     Title
9588 BN  1532 ALL    W3VH    W3IWI  851122 IBM WØRLI Clone
Via W3VH From W3VH: Msg# 238, Rcvd 851122/Ø321z, Sent 851122/Ø325z
 233 BN  14Ø7 ALL    W3VH           851121 IBM WØRLI Clone
```

The following ported from Compuserve Hamnet:
I have almost finished a translation of the WØRLI BBS software into Turbo Pascal for use on IBM PC/XT/AT. The Mail portion, including forwarding, and the Gateway portion all seem to work fine. I havent added "J" lists or monitoring yet, and still have to add a few local commands. I will upload the program as soon as it is finished. However, there is very little activity here (at present I can contact 2 others on Packet) which makes testing a little difficult. Also, I am working from a source listing, but have never seen a "real" WØRLI BBS in operation and have no access to an 82Ø to try it, so I cant be sure if it looks exactly correct. If someone is interested in testing it as is, I would be gad to mail acopy on diskette before I spend a small fortune uploading it here. I am working with the version 1Ø.2 source and should be compatible with the latest forwarding methods. If you really want to test it, send me your address via Easyplex (so it doesnt scroll away while I'm at COMDEX) and I'll mail you a diskette. If you just want to try it, mail me a diskette and I'll copy it off for you. I'm testing it as best I can here with a TNC-1 on one port and a TNC-2 (v1) on the oter with no apparent problems.
Jeff Jacobsen, WA7MBL, 14ØØ E 9ØØ N, Logan, UT 84321
MCI Mail: 23Ø-9237   Easylink: 62845849   BITNET: BOBW@USU CS: 72446,2557

M 169ØØ ROBERT J DIERSING (N5AHD,2959) 11/22/85 9:54 PM L:5Ø
KEYS:/PACKETS AT PROGRAMMING CONTEST/
TO: (Group 95)

ACM South Central Region Programming Contest Problem 1

A text message has been received from the ACMSTAT satellite. The message is encoded as a file of packets named packets.dat. Each packet contains seven (7) characters along with error-detection and correction information. You are to create files message.dat and errors.dat which are to contain the decoded message and an error report, respectively. These files are described below.

- - - - - - - - - - - - - -

The above is the preamble from one of the programming problems for the South Central Region ACM programming contest held in Houston last weekend in

conjunction with the annual regional meeting. I surely didn't expect to find a reference to packets and satellites in any of the problems and I though that the group might be interested in this. There was also a problem involving finding reachable nodes in a computer network.

An interesting side story is the fact that I have been teaching a telecommunications course this semester using "Computer Networks" by Tanenbaum. Those of you familar with this book know that Chapter 6 is devoted to satellite and packet radio networks. Needless to say, a demonstration of several TNCs was done at the appropriate time in the course. The only holder of an amateur license in the class is not active on packet but he was a member of out contest team. He found it quite amusing when few of the other teams really had a handle on the problem.

Now I'll have to blow our horn a little. Most people have never heard of Corpus Christi State University but you will have heard of some of the schools in the overall ranking of teams: (1) Rice #1, (2) Texas Tech, (3) Rice #2, (4) CCSU, (5) Texas A&M #1, (6) University of Texas at San Antonio, (7) University of Texas at Arlington #1, (8) UTA #2, (9) Tulane, (10) Baylor, (11) Texas Christian University #1, (12) SE Louisiana, (13) University of Texas at Austin #1, (14) University of Southwestern Louisiana #1, (15) Texas A&M #2, (16) University of Tulsa, (17) USL #2, (18) TCU #2, (19) McNeese, (20) University of Arkansas, (21) East Texas Statee, (22) Lamar, (23) University of Houston - University Park, (24) Midwestern, (25) UT Autsin #2, (26) University of Houston - Clearlake, (27) Oklahoma State #2, (28) Oklahoma State #1, (29) SLU Hammond.

Our team was in division I is which teams consist of at least one graduate student. In division II all team members are under- graduate students. As far as we are concerned, a 4th place in division I is nothing to sneeze at. In the combined standings they were 5th out of 45 teams.

73, Bob

C2987 CC236 Tom Clark (W3IWI,2976) 11/22/85 2:08 AM L:46
KEYS:/DE WB6RQN/TNC2 PROBLEM/HELP SOUGHT/

The following is relayed from the W3IWI BBS:

```
Msg# TR  Size  To     From    @ BBS   Date     Title
9573  Y  1653 W3IWI  WB6RQN          851121 TNC2 problems
```

Tom please forward to the appropriate folks (thanks).
Subject: TNC2 problems
I have been having a rash of problems with my TNC2 that brings its reliability into question. First the environment.
My TNC2 is connected to a UNIX SysV system allowing hams to enter and use my UNIX machine. When a connection is established, DCD from the TNC goes on, and UNIX presents the login message. This works -*MOST*- of the time. The TNC2 is a Rev 1 with the 1.1.0 firmware. The TNC is all CMOS with the exception of the SIO. The TNC is placed in the transparent mode and remains there always. TXFLOW and TRFLOW are enabled and seem to be working.
The TNC fails in one of several manners:

1) The TNC will ignore all connect requests but will digipeat frames and will continue to beacon.
2) The TNC will ignore connect requests when the requesting station is coming through a digipeater, but will readily connect when the station is direct (no digis).
3) The TNC will connect but will not pass any information to the computer. Connecting a terminal will allow me to issue commands, connect, and disconnect but I cannot ever get it to pass data in the direction of the terminal or computer. When it gets wedged in this mode the only remedy is to power down the RAM (remove it from its socket). Turning the power off then on doesn't help in this case, although it fixes #1 and 2 above.

I am at Witts End. I do not know how to force the TNC into this mode although circumstances get it there about once every two days.
73 de Brian, WB6RQN

BBS: WB6RQN @ W3IWI
uucp: ...!bellcore!yojnal!wb6rqn
N4KRR would like to use the MailBox when you are done, Tom.
W3IWI at 851121/1426: Last msg # 9577, 72 active msgs
(B,D,G,H,I,J,K,L,R,S,T,U,W,X) >
Msg# TR Size To   From   @ BBS   Date   Title
9577 N   438 WB6RQN W3IWI   851121 TNC2 problems
Brian -- one thing I discovered here that caused some grief that might be
ur problem. Look at the manual under CANPAC and notice that if a CANPAC
character is received in cmd: mode, all output ceases until a second one
is rcvd. Here I have disabled CANPAC (and also CANLINE for safety) by
setting both to zero. I got burned when a bad spot on a disk got a ctl-c
and then later a ctl-y embedded in it inadvertently. See if that fixes it.
W3IWI at 851121/1427: Last msg # 9577, 72 active msgs
(B,D,G,H,I,J,K,L,R,S,T,U,W,X) >

C2987 CC237 Howard Goldstein (N2WX,2987) 11/22/85   8:42 AM L:24
KEYS:/LEVEL 3/VERY PRELIMINARY CODE/+READ/USE AT OWN RISK!/

To: Curious Beta
Fm: Howie N2WX
Re: AX.25 Level 3 for 16K TNC 2s
dist: Closed (it ain't done)

Hi folks I have placed here on EIES some .HEX (quite a bit actually) that runs level 3.

IT IS VERY PRELIMINARY AND VULNERABLE TO ABUSE! I offer it merely to placate any disbelievers you may have run into who say it can't be done on a z-8Ø or it's too hard to implement or any number of other specious excuses.

Please +READ C2987 CC237 to get about 2 pages of survival sheet, and then the appxly 59K of .HEX. I suggest that if your interested, do the +READ but interrupt it when it starts on the .HEX. Peruse the "documentation" and decide if you want to +READ the whole thing.

Obviously to get full benefit (THAT claim is specious in its own right at this time!) one should have access to at least two TNC 2's.

73 Howie

C2987 CC238 Phil R. Karn (ka9q,2979) 11/22/85   2:57 PM L:15
KEYS:/ME TOO!/

My TCP/IP implementation has been running nicely on the IBM PC for the past week. It has been tested successfully with about 8 or 9 different TCP implementations out on the ARPA Internet. Tests included sites like Raisting, West Germany (whoever said that TCP/IP was forbidden in Europe?) and self-connects through special "echo" servers.

HINT:
If somebody will give me a programmer's view of the TNC-2 (port assignments, memory layouts, etc) I can begin porting this code, which is entirely in C, to the TNC-2.

By the way, the total size of the runnable program for remote login is 18K on an 8Ø88, and this includes quite a few K of debug "scaffolding".

Phil

M 17Ø1Ø Rich Amundson (WAØJFS,2965) 11/23/85   9:11 PM L:22
KEYS:/SAREX-2/
TO: (Group 95)

TO: MAPR, SLAPR, OTHER MIDWEST GROUPS

FROM: RICH AMUNDSON   CITS
RE: SAREX-2

IN TRYING TO HELP DEBUG AND TRY OUT THE SAREX-2 HARDWARE SYSTEMS, CITS WILL BE A BETA SITE OF THE SOFTWARE. ONE OF THE TRIALS WE WILL USE WILL BE TO PUT A COPY OF THE SYSTEM IN AN AIRCRAFT AND ORBIT AT ABOUT 10,000 FEET OVER CENTRAL IOWA. WE HAVE DONE THIS BEFORE IN CONJUNCTION WITH THE NORTH AMERICAN TELECONFERENCE NET AND HAVE HAD RESPONSES FROM THE KANSAS CITY, ST. LOUIS MINNEAPOLIS, CHICAGO AND CENTRAL NEBRASKA AREA. WE WOULD VERY MUCH APPRECIATE IF ANY OF THESE GROUPS OR INDIVIDUALS IN THIS ARE WOULD LIKE TO CONTRIBUTE FUNDS TO OFFSET THE COSTS OF THE AIRPLANE. WE ARE USING A CESSNA 182 THAT NØBKB HAS AN INTEREST IN. THE COST WILL BE ABOUT $38.00 PER HOUR I BELIEVE. WE WILL BE USING ABOUT 3 HOURS OF FLIGHT TIME TO GET A GOOD SOLID 2 HOURS ON STATION WHICH SHOULD GIVE A GOOD INDICATION OF HARDWARE CAPACITY. WE WILL ALSO BE CAPTURING THE DATA DOWNLINKED TO USE IN THE EFFORT TO GET STATION HEARD AND CONNECT LISTS.

IF ANYONE HAS SOME HELP TO GIVE PLEASE LEAVE ME A NOTE HERE OR YOU CAN GIVE ME A CALL AT 515-285-3352 EVENINGS. THANK YOU.

73'S   RICH   WAØJFS

C2973 CC44 Rich Amundson (WAØJFS,2965) 11/23/85   8:57 PM L:16
KEYS:/NETWORK CONTROLLERS/

TO: LYLE JOHNSON(TAPR)
FROM: RICH AMUNDSON(CITS)

LYLE,

DAVE HUFMAN, ON SEEING ALL RELEASED DATA ON THE TAPR NNC, DOESN'T SEE ANY REASON THE WORK WE ARE DOING WON'T PORT OVER TO THE 64180 ENVIRONMENT AS IT IS IN Z-80 ASSEMBLER. WE ARE AT THE POINT OF APPLYING POWER TO THE 9600 MODEMS TO SEE HOW THEY MARRY TO THE FM-5 BOARDS.
THEN WE SEEM TO BE IN THE MIDST OF THE SAREX-2 PROJECT ALSO WITH MYSELF AND NØBKB PLANNING TO MAKE A FLIGHT OF THE HARDWARE WITH WA2GTM AS OPERATOR ON DEC 11 OR THE 18TH FOR WEATHER BACK-UP. THIS SHOULD GIVE A GOOD SHAKEDOWN OF THE SYSTEM BASED ON THE RESPONSES WE GOT WITH SIMILAR FLIGHTS WE MADE FOR TWO OF THE TELECONFERENCING NETS LAST YEAR.
KEEP UP THE GOOD WORK.
73'S   RICH   WAØJFS

M 17130 J. Gordon Beattie Jr. (N2DSY,2990) 11/24/85   5:14 PM L:10
KEYS:/IS A CESSNA 414 FOR FREE USEFUL TO THE SAREX-2 TESTING ?/
A: 17010   TO: (Group 95)

Rich et al:

Regarding the SAREX-2 test flights, I travel each week for a total of thirty hours in a Cessna 414 at altitudes ranging from 8500 to 17,000 feet. Routes include New York to Texas, Alabama, Missouri, Illinois etc. Can we be of assistance in testing of the package ? We can provide pressurized or non-pressurized environments. I will be able to post airtimes a week or more in advance. If this is useful let me or Tom know.
73, Gordon

M 17147 Lyle Johnson (WA7GXD,2973) 11/24/85   9:27 PM L:27
KEYS:/NNC/DRNET/IMAGE/PROJECT/WIDEST POSSIBLE DISSEMINATION/
TO: WA7GXD, (Group 95)

all,

I have been speaking to a number of people recently, and there have been a number of negative comments about DRNET and "secret projects" -- both in reference to the NNC project.

PLEASE PLEASE PLEASE PLEASE PLEASE if you haven't already done it, check into conference 2974 and 2973 and put the info there on the NNC discussions on your PBBSs for wide dissemination.

discussions on your PBBS for wide dissemination.

Remember, DRNET is for info spreading throughotu the packet community, NOT a private messaging system. The NNC is NOT a "secret project" and I have been actively soliciting inputs from the widest possible audience.

I am happy to report that the SouthNet II conference in Atlanta, from which I returned a couple of hours ago, was a very interesting and successful regional packet forum. The NNC was publicly shown for the first time, and there is a lot of positive response. I will be porting on DRNET, FOR WIDEST POSSIBLE DISTRIBUTION< a description of how to get involved in working on the project and testing it in the next day or so.

Again, if you have a PBBS, or easy access to one, PLEASE post these discussions on the NNC. Reaction to the NNC has been generally very positive, but we have an image problem with DRNET being a closed medium. That is NOT the intent.

Thank you. level three, here we come!

M 17541 Gwyn Reedy (W1BEL,2975) 11/26/85 6:01 AM L:29
KEYS:/CALL FOR PSR ARTICLES/
TO: (Group 95), W1BEL

The deadline for the January issue of the PSR Quarterly is Dec 20th. This is a 'no-slip' date because the PSR has to be finalized and in the Fed-EX to TAPR by DEC 30 in order to get the director ballots out in time before the board meeting.

.

I encourage all previous and potential writers to contribute. The support to date has been fantastic, and I am hopeful it will continue at such a strong level. Articles about construction, modification, or understanding of hardware and software are especially useful. As always, more info on networking is desired, either tutorials, or cannonballs in the 'protocol wars.' Historical articles would be nice too - remember that half of the packeteers in existence got on the air this year, and an old timer is a guy that was on in 1984...It is good to document things that went on in 1978 - 1984 for these people.

.

Please send in those articles. I will do the minimum amount of editing to your input. Primary editing is done to reformat and reduce length in order to improve layout. Send here, or CIS (76576,2003), or mail a diskette (8 inch SSSD or 5 incy Kaypro 4, or 360k PC format), or as a last resort, a typed copy. Mail to PSR Editor, 812 Childers Loop, Brandon, FL 33511. If there are diagrams or drawings, either send camera ready copy or get the drawings in extra early so there is time to have them redone.

.

Thanks and I'm anxiously awaiting inputs. Gwyn

.

P.S. Good reviews of the TNC2, Kantronics rev 1.6, AEA PK-64 and PK-80, the new GLB low power model, new terminal software etc, would be greatly appreciated by the readership.

M 18103 Lyle Johnson (WA7GXD,2973) 11/28/85 11:35 PM L:87
KEYS:/THE RUMORS ARE TRUE/PLEASE SPREAD THE WORD/
TO: WA7GXD, (Group 95)

To: All Amateur Packeteers
Fm: TAPR
Re: Availability of TNC 2 (or, The Rumors are True!)

For Immediate Release

Tucson Amateur Packet Radio is pleased to announce the cessation of its production of the popular TNC 2 kit!

Yes, pleased. An all volunteer R&D, manufacturing, marketing, and support organization is fun for awhile, but it soon takes a

great toll on all involved.

TAPR began life in 1981 with a simple goal: To experiment with packet radio and packet radio networks. At that time there was little in the way of easily acquired packet equipment. TAPR took a short side trip into the manufacturing world - to permit widespread experimentation with packet radio by making high quality software and hardware building blocks available on a large scale at low cost.

That job is now complete. The TNC 2 design is stable, and presents a good balance between small size, low power, and ease of manufacture, while still allowing for high speed, full duplex operation, adaptability to future requirements such as split baud rate operation, and easy interface to future modem designs.

It is now time to turn the manufacturing and marketing tasks over to industry, and to continue with Research and Development in new areas of packet radio. We have made our TNC 2 design available to others through an OEM agreement. These alternate sources are listed below:

Advanced Electronic Applications, Inc.
PO Box C-2160
Lynnwood WA 98036
(206) 775-7373

Model PK-80. Assembled and tested only. Available from dealers. (AEA also produces the PKT-1 -- a TAPR TNC 1 clone -- and the PK-64 designed especially for the Commodore 64 computer.)

GLB Electronics, Inc.
151 Commerce Parkway
Buffalo NY 14224
(716) 675-6740

Model TNC-2A. Kit form only. Factory direct. (GLB also produces the PK-1 and PK-1L "software approach" assembled and tested TNCs.)

MFJ Enterprises, Inc.
PO Box 494
Mississippi State MS 39762
(601) 323-5869

Model MFJ-1270. Assembled and tested only. Available from dealers.

PacComm Packet Radio Systems, Inc.
4040 Kennedy Blvd., Suite 620
Tampa FL 33609
(800) 835-2246 Ext. 115 (orders only)
(813) 689-3523 (technical/service info)

Model TNC-200. Assembled and tested/Complete kits/Partial kits. Factory direct.

Therefore, assured of the continued availability of high quality TNCs at reasonable prices, TAPR has ceased production with TNC 2 order #1200 (s/n 3721).

TAPR will continue to support software development for the TNC 2 as well as the earlier TNC 1. Other ongoing projects include the TAPR NNC (a networking controller), higher-speed modems, packet

satellite support, and other areas of packet development. If enough demand exists, TAPR may make bare boards (no parts) available to experimenters. TAPR has no new TNC hardware products in development at this time.

We invite you to become a member of TAPR, and to attend the next annual membership meeting in Tucson on February 8th, 1986.

TAPR wishes to extend its heartfelt thanks to you, the Amateur community, for your strong and continuing support.

M 18522 Tom Clark (W3IWI,2976) 12/ 1/85   2:ØØ AM L:3
KEYS:/TNC2/C1Ø REPLACEMENT PART/TRY RADIO SHACK/
TO: (Group 95)

For those of you looking for capacitors to replace C1Ø on the TNC2, you might want to check Radio Shack Part # 272-1Ø27, 5Ø uf @ 35V, which fits the board like it was made for it. The price is $Ø.69

M 18523 Tom Clark (W3IWI,2976) 12/ 1/85   2:Ø7 AM L:17
KEYS:/BUG REPORT/KANTRONICS REV 1.6 SOFTWARE/AEA PK-64 TNC/
TO: (Group 95)

This is an inquiry to see if any of you have noted the following anomalies in the Kantronics and AEA PK-64 TNC's:

Kantronics with rev 1.6 software has been observed to go into a mode we have humorously dubbed "stuttering". A packe is sent by the Kantronics, acked by the receiving station, and then resent exactly the same, something like

```
   This is a test 1234         (packet sent and acked)
   This is a test 1234         (packet is re-sent again, despite ack)
```

AEA PK-64: twice we have seen a packet with FRMR errors sent by PK-64, both times when connected to an 'RLI BBS. In one case it was sent to me where it was detected by a TNC2, and in the other case it was detected by W3VH using a Kantronics. The BBS sorta chokes when this happens 'cuz it doesn't really know how to cope with a FRMR error message!

73, Tom

C2973 CC45 Phil R. Karn (ka9q,2979) 11/24/85   8:Ø4 PM L:168
KEYS:/SOME THOUGHTS FROM W3HCF/

Date: 24-Nov-85 Ø5:23:37-UT
From: mills@dcn6.arpa
Subject: Essay for the Digital Committee
To: packet-radio@mit-eddie.arpa

Folks,

There have been many informal discussions recently on the issue of "level-three" or "long-haul" protocols. The ARRL is planning meetings which will both create and solve problems in this areas. Phil Karn is trying to drum up support for these meetings in general and for IP/TCP in particular. Like some of the rest of us, Phil has learned a few lessons actually building and using high-flake networks that bear a strong resemblance to those likely to be built by the amateur community. On the other hand, there is a stong lobby, including mostly those who have built low-flake commercial networks and those who have built no networks at all, who display incandescant X.25/X.75/X.121 bumper stickers.

Personally, I think that conflict between the two lobbies is silly and that both communities can co-exist quite handily, as long as care is taken in the architecture. The best existence test might be the DARPA Internet, which includes X.25 nets, Ethernets and all kinds of others, even AX.25 nets, right

now. However, I am worried that some zealot might cop off something silly at one of these meetings and that may result in shutting down some very useful options.

Accordingly, I herewith break my promise to confine mutterings to my own swamp and offer the following essay. You should understand the opinions are definitely my own and that the exposition is intentionally tart. My hope is to get you all mad in the first five minutes after reading it, assuming you survive that trip. Please do not compose a reply during that interval. Then, I hope that you will think through the issues and direct buttals and rebuttals not at me, but at the members of the ARRL Digital Committee.

On the Standards Process

Research tends to expand the number of options, while standards tends to decrease them. One should push on either one until the rate of increase equals the rate of decrease and so that the total number results in a healthy and vigorous community.

Amateur radio is after all a recreational community, a fact which some of our more agressive special-interest groups sometimes forget. As technology matures standards often work to encourage new entrants and to stabilize exploitation, with the amateur satellite community being a good example. Pre-emptive efforts to standardize, especially in a recreational community, would work to discourage innovation, enthusiasm and participation in the standards process itself. Volunteer standards (e.g. bandplans) work only if there is a clearly defined benefit which extends beyond the boundaries of the special-interest group. In other words, resist telling someone what's good for him unless he asks.

Any standards process that does not represent the consensus of a constituent majority or is opposed by a significant minority who may be bound by it will be unpopular, discredited or ignored, regardless of technical merit. Any standards process that attempts to restrict the scope of design, implementation and experiment must be thoroughly justified on engineering grounds. Political justifications are unacceptable. The notion that our community can't do something because the Europeans (or the Martians for that matter) won't accept it is too preposterous for comment.

Every standard has a definite life cycle from the point at which a need for it is identified, through the coordination and specification process, prototype implementation and testing, then a period of refinement and evolution until finally it is overtaken by a successor. A standard must never attempt to stifle the evolution of successors; in fact, it must encourage it. One of the implications is that every packet header must include a protocol identifier and version number, so that different protocols and versions can co-exist in the same system.

Any network-level standards proposed for use in the amateur packet-radio community must address the issue of the radio channel a-priori. An "amateur" standard requiring, for instance, use of X.25 or SLIP on wire links between local-area gateways, is completely out of scope. Such would not be an amateur packet-radio standard, merely a bilateral agreement between the gateway operators. The amateur community is best served by concentrating on the engineering problems and their solutions with respect to radio issues and only secondarily on the more well-studied non-radio issues. The Manufacturing Automation Protocol suite is a good example of how to resist the urge to reinvent the wheel.

General Assumptions

While the existing CSMA/FM technology has little going for it on pragmatic engineering grounds, its widespread use is a fact. The AX.25 frame-level procedures are certainly useful, but badly matched to the CSMA/FM channel characteristics. Much can be done to improve operational procedures to reduce congestion and improve throughput, including the use of multiple channels and

cross-channel bridges. However, development of the long-haul system must not presuppose such improvements.

The existing organization can be described as a datagram network with stateless repeaters and reliablity provided by end-end resequencing and retransmission. It may happen that some long-haul systems will include internal provisions for resequencing and retransmission in order to improve performance. However, such provisions must not obsolete existing end systems and protocols. Specifically, this requires an overall system specification on maximum permissable delay.

Datagram routing between end systems is presently determined by an explicit indication of source route in every frame. Pragmatically, this can be justified by the fact it works reasonably well in local-area systems where there are limited choices of available routes and the number of hops is small. It may happen that sophisticated routing algorithms will be developed for use in some local areas; however, a long-haul system must not presuppose this.

Development of the long-haul system must not depend upon a particular choice in addressing or routing semantics used in a local area and must not require changes in existing local-area procedures when working other stations in the local area. This implies the long-haul system must be able to interconnect existing Vancouver and AX.25 communities and possibly others that may come along, even if they share the same radio channel.

On Datagrams vs Virtual Circuits

Experience in the education, research and development community suggests the use of an internetworking protocol and addressing paradigm based on either the DARPA or ISO connectionless protocol suites. In this model end-end connectionless (datagram) service is the primary communication mechanism, with reliability achieved through the use of end-end internet virtual circuits, perhaps supplemented by intranet virtual circuits. Gateways between connectionless networks are stateless and operate only in datagram mode, so are relatively simple.

Experience in the common-carrier community suggests the use of an internetworking protocol based on the X.75/X.121 protocol suite. In this model connection (virtual-circuit) service is the primary communication mechanism, with reliability achieved through the use of concatenated virtual circuits between the gateways. Gateways between connection networks must preserve state for each individual end-end virtual circuit, so are relatively complex.

Given the present flux in design philosophy and opinion, it is inappropriate for the views of either community to dominate the standards process. The standardization, design and implementation of long-haul amateur packet-radio systems should not preclude access from either communty to the other, perhaps via strategically located protocol-translation gateways. Furthermore, the basic long-haul transport mechanism must provide for both connectionless and connection service on an end-end basis. Many examples of how to do this are apparent in the DARPA Internet system.

On Interoperability

There are now and will be in the future numerous opportunities to proliferate access to and between amateur packet-radio systems via public and private nets including, but not limited to, the DARPA Internet, VAN nets (Telenet, etc.), BITNET, USENET, MAILNET, CSNET and others. In many cases the access must by controlled on economic or policy grounds. This is a very tricky issue in view of the individual policies of these nets and the government regulations under which the Amateur Service operates. In fact, the duly approved use of one or another of these nets or even ordinary dial-up modem connections may be a useful alternative to a strictly Amateur-Service backbone trunk in some cases.

The most important factor in these issues may be the need for survivability and interoperability in case of civil emergency or disaster communications

and interoperability in case of civil emergency or disaster communications.
The standards process must consider these implications with respect to interconnecting to other link technologies, for instance, an HF link as backup for a 220-MHz multiple-hop backbone or an XMODEM or KERMIT dial-up link as a bridge between mail forwarders.

Note very carefully that these factors must not be allowed to advance the cause of any particular protocol, such as X.25, Internet or XMODEM, without considering the entire protocol suite of which this protocol might be a component. It makes no sense to blindly speak X.25 on the basis that "everybody does it" without considering what applications above X.25 that the emergency-communications community will need and have available.

Dave

C2973 CC46 Phil R. Karn (ka9q,2979) 11/30/85 10:25 PM L:17
KEYS:/CURRENT TCP HAPPENINGS/

In case anybody's interested (Jack, are you out there?) someone has ported the MIT TCP/IP package originally written for the IBM PC over to the MacIntosh. I understand this is in the public domain, since it was posted on USENET. I'll fetch it and figure out how to get it to anyone who's interested.

My own code has now been tested on several processors, including representatives from the Big Endian family (68K) as well as Little Endian (8088) in order to flush out any byteordering bugs. The TCP is stable. Recent additions include a common "socket-like" programming interface (net_open, net_send, net_recv, net_close) and support for the User Datagram Protocol (UDP). UDP is not yet tested. Several others on the East Coast are currently active in porting my TCP to several other machines, including the Commodore Amiga. More as it develops.

Phil

C2974 CC103 Skip Hansen (WB6YMH,2964) 11/24/85 1:49 PM L:29
KEYS:/9600 BAUD STUFF/

Two K9NG modems are mostly on the air in Southern California. We have exchanged packets between Brea and P.V. about a 40 mile path using midland 13-509's on 220.95. The modems fired up on the bench the first time. A few problems were noticed however:

1. Midland 13-509's hear themselves on 220.95! Critical retuning of the oscillator cures this problem. (or just puts it off maybe ?)

2. 220.96 seems to have voice traffic on it. (boy never ran into this before have we?)

3. The PTT and DCD Led drivers on the TNC-1 are from before the modem disconnect connector so they are inoperative when the 9600 baud modem is plugged in. (This was easily fixed with an Exacto knife.)

4. The DCD circuitry thinks our squelch noise is carrier most of the time. Changing C18 from a .047 to a .1, removing R30 and D3, and jumpering the emitter of Q6 to C18 helps greatly. At this point DCD falses about 2 to 3 times a second for a couple of milliseconds. I am not at all happy with this mod, but it does allow the transmitters to transmit.

We are in the currently in the process of replacing the Midlands mechnical TR relay with PIN diodes and transistor switching of the receiver and transmitter voltages.

Does anyone have any ideas on the DCD problem ?

(it usually doesn't! we will want to put ...

73's Skip WB6YMH

C2974 CC1Ø4 Skip Hansen (WB6YMH,2964) 11/3Ø/85   6:54 AM L:28
KEYS:/MORE 96ØØ/

Well the problem I reported in an earlier message with the DCD output turned out to be the deemphasis circuitry in the receiver rolling off the high frequencies too much. After I removed the deemphasis cap the DCD stopped falsing. And we connected successfully! The error rate seems good, at least 8Ø % of the packets are getting there on the first try. We are still using the relay switching on the Midland's and hope to be getting some PIN diodes to play with soon. I am using an coat hanger ground plane inside of the shack with 1Ø watts talking to Brea (about a 4Ø mile path).

Midland 13-5Ø9 mods so far:

1. Cut C42 demphasis capacitor out of the circuit.
2. Take the Disc. feed for modem from the junction of R26, C41 and C39.
3. Cut R11Ø to remove the phase modulator input from the transmitter.
4. Remove one of the fixed capacitors which is in parallel with the transmitter trimmer capacitors and replace with a MV2111 varicap.
5. Connect a 1Øk resistor from the modem output to the hot side of the varicap.
6. Bypass the junction of the 1Øk and the modem output to ground with a .ØØ1 cap as close as possible to the resistor.
7. For a keyed to ground PTT signal remove Q2 from the modem and jumper the collector of Q3 to PTT. (This is just a temporary stop gap to test the setup while we get the PIN diode switching built).

Well WB6YMH-1 and WB6KAJ-1 are on the air at 96ØØ, where's the crowd ?

73's Skip WB6YMH

C2974 CC1Ø5 Lyle Johnson (WA7GXD,2973) 11/3Ø/85   8:23 PM L:139
KEYS:/NNC SOFTWARE DEVELOPMENT/PLEASE SPREAD THE WORD/HELP WANTED/

To: All Interested Packeteers
Fm: TAPR
Re: NNC Development and Testing

As many of you are already aware, the new TAPR Networking Node Controller (NNC) is nearing completion of prototype hardware debugging. The NNC is a four-port packet controller with a large memory area, direct-memory access (DMA) capability for I/O -- and Z-8Ø software compatibility.

To dispell any rumors, there presently exists NO SOFTWARE for this device. None. Nada. Ayn. Zip. Zero. Effes. Klum. {   }

What we are looking for are volunteers to assist in developing software for this device.

We need low-level, highly-efficient drivers for the I/O. We need an AX.25 Level Two handler that can handle multiple logical and physical channels. We need Level Three and Level Four. We need loaders for uploading software updates to a remotely-sited NNC. We are hoping that there will be early porting of multi-port digipeater code to this unit as well as a WØRLI PBBS. We need close coordination of the various aspects of the development. We need... You get the idea.

The hardware should be verified during December. If all goes well (it usually doesn't) we will want to put Alpha units in the hands

(it usually doesn't), we will want to put Alpha units [illegible])
of developers in late January/early February. Assuming a couple of months to get enough software together to make Beta testing meaningful, we will be looking for Beta testers in the March-April timeframe. Once testing has advanced to the point of reasonable confidence, we will make the units generally available (summer of 1986?).

Now, we are NOT looking for folks who want to be the first kid on the block with a new toy. We need people who are committed to Amateur packet radio and want to help make a meaningful contribution to a very large and difficult task.

And be forewarned. You may slave away for many, many hours, only to have your code not used, or superceded, or... No guarantees.

Coordination is going to be a tough assignment. Without proper coordination, a lot of wheels will spin, and a lot of energy wasted in duplication of efforts. A BBS to swap code modules will be needed. All code will need to be carefully, accurately and exhaustively documented -- by the author!

Developers will need to procure the following:

1) One NNC digital unit - projected cost is $175. This is an NNC with uP, 64k bytes of bbRAM, 32k bytes of EPROM, four HDLC ports, two parallel (centronics compatible) ports, two async ports and one SCSI interface. The SCSI chip may not be included at this price, we are not sure yet, but for the Alpha testers/developers it will be. This unit will be fully assembled and "tested."

2) One NNC Floppy Adapter - projected cost $125. This includes a DMA'ed Floppy Controller that can handle 4 diskette drives. This unit will NOT support 8" drives (lack of 8" support is intentional). The price includes a licensed copy of Z-DOS, a CP/M 2.2 compatible operating system. It will be on 5.25" double-sided 48 tpi diskette format capable of 368k bytes (formatted). If the decision is made up front to use 96 tpi drives, TAPR will copy the licensed diskette to the denser format and supply the original as well as the high-density copy to the purchaser.

3) A pair of 5.25" floppy drives. Maybe we can do a group purchase of TEAC 55Bs. Figure $15Ø for this expense. 48tpi or 96tpi are about the same price. 48 tpi yield about 4ØØk formatted bytes; 96 tpi about 8ØØk formatted bytes.

4) A power supply. $5Ø from surplus sam?

5) One NNC Modem board - projected cost is $15Ø. This is a wired and tested board which includes one 3ØØ-baud 22Ø6/2211 modem with tuning indicator and three 12ØØ-baud 22Ø6/2211 modems. We might get this cost down to $125.

Thus, there is a cost of participation that will be a minimum of $45Ø and may be $65Ø. Add to this the cost of an assembler or compiler...

The assembler that seems to make the most sense is ZAS, from Echelon systems. Again, we can probably do a group purchase or multiple-site license for this project. This assembler supports the extended instruction set of the HD6418Ø cpu. There is no reason to limit ourselves to the Z8Ø instruction set (or - yeech - the 8Ø8Ø subset) for this project. And ZAS is fairly cheap - about $5Ø, I think.

I don't know which C or Pascal compiler will be chosen. I assume that one will be chosen, so the high-level code can be written in a transportable high-level language (makes for easier testing?) while the interfaces to the hardware can be done in assembly language. Preferably, the compiler will generate Z80 (or 6418Ø) source code for assembly by ZAS. This allows hand-optimization of the compiler output.

current input stream.

By standardizing on the development environment (NNC w/5.25" floppies) and the tools (assembler, compiler(s)), we hope to make it easier for all participants to share their work amongst the group.

It is expected that all code (including source code) developed for this project will be placed in the public domain for non-commercial use. And that TAPR will be given explicit (not exclusive) right to distribute it.

If you have the time and ability and want the chance to make a real contribution to Amateur packet radio networking development, please leave a message on DRNET or write the TAPR office. We will put you on file and notify you when we are ready to get started with Alpha test or Beta test (as you indicate to us).

For Alpha test, we need developers. Committed developers. People who really understand software design, hardware/software interaction, protocol implementation, code size/speed tradeoffs, data structures and myriad other facets of software design. And of course, understand networking...

For Beta test we need testers. People who are in a real packet environment, who have a good site that will get plenty of exercise on the air, who have the time and commitment to submit detailed reports of what works and what doesn't. This isn't a "be the first person on your block to own an NNC" contest; it is going to require work.

What do we mean by commitment?

Consider a plate of bacon and eggs. The chicken was actively involved; the pig was committed.

If you are a capable packeteer, committed to assist in networking development, please provide us with the following:

Full Name.
Amateur Callsign.
Mailing Address.
Daytime telephone number.
Evening telephone number.
Alpha or Beta test.
TAPR membership number (if applicable).
Specific areas of expertise that you wish to make available to
  this project (low level interface/high level protocol implementation/
  documentation/testing/etc.)

Thank you for you help. Happy packeting!

C2987 CC239 Howard Goldstein (N2WX,2987) 11/26/85 3:17 PM L:3Ø
KEYS:/RELEASE 1.1.1J AVAILABLE/+READ THIS ITEM/FOR 16K RAM ONLY/

To: Tiny
Fm: Howie N2WX
Re: 1.1.1j ($ØB checksum)
Dist: Beta testers

Please +READ C2987 CC239 to get .HEX for 1.1.1j.
CHANGES:
o- Headers no longer displayed from connectee while MCON and MON ON
o- Transparent echo no longer occurs
o- LCOK, ESC repaired
o- BHeard removed, got plenty of negative comments on that one!
o- NULL handling totally revamped, no longer are nulls buffered
o- New command LCStream (default ON) performs a "toupper()" on stream selector the user issues
o- Strange, seemingly random happenings to link state hopefully corrected.
o- CMSG and CTEXT handling changed: before, CTEXT would only go out on the current input stream. Now CTEXT goes out on the newly connected stream SO LONG

[illegible]
AS
STREAMSWitch is NOT Ø !!! (and of course, CMSG is ON)
 o- and finally, true break escape to 'cmd:' mode from transparent works better now, doesn't send out the packet with a single Øx00 byte
 -----
 A final note, this release hasn't been thrashed at 9600 baud yet so please do do thrash it if your capable and report results back to the conference!
73 Howie

C2987 CC240 Howard Goldstein (N2WX,2987) 11/28/85 10:16 AM L:26
KEYS:/PORTS/MEMORY/ASSIGNMENTS/CAVEATS!/

```
to:        All (Paul pls correct if I messed up)
fm:        Howie N2WX
re:        Port, memory assignments TNC 2
dist:      open
```

RAM and EPROM on the TNC are mapped as thus:

EPROM starts at Øx0000 and goes up
bbRAM starts at ØxFFFF and goes down

On reset or power on clear the PC is loaded with Øx0000

Ports:

```
SIO A dat      ØxDC
SIO A ctl      ØxDD
SIO B dat      ØxDE
SIO B ctl      ØxDF
```

Hardware signal driving SIO syncB* is 600hz symmetric

C2987 CC241 Tom Clark (W3IWI,2976) 11/30/85 2:34 AM L:10
KEYS:/HOWIE/RE:LATEST CODE/A FEW QUESTIONS/
A: 239

Howie, downloaded the HEX for the latest software rev and noticed that you exactly filled it to the 16k boundary (according to the last line of the hex dump). Is it possible that the length is exactly 16k or is there some code missing. I haven't tried burning it and testing it yet. Sorry you pulled BHEARD Y|N out -- the folks around here like it! Re the VC X.121 code, in order to get someone else to try it with, I'm going to have to "break" the sanctum sanctorum of the "beta" group. Any objections if I make it available to a few selected others locally for testing?

73, Tom

C2987 CC242 Howard Goldstein (N2WX,2987) 11/30/85 8:22 AM L:12
KEYS:/1.1.1J SIZE/BHEARD/SAREX2/NON BETA NETWORK TEST/
A: 241

Tom - Right 1.1.1j fills a 27128 or 2 64s exactly to the limit no problem there.

BHeard is gone...but not forgotten! The SAREX2 version does it but it may not have been suitable for general use it could encourage every jack in a mineshaft to do a heard beacon thru 8 digis.

I don't have any big problem with the network release being tried by non-betites provided they know it ain't done and there are known problems. With that in mind though I'd be pleased to have it thrashed about!

73 Howie

C2987 CC243 Tom Clark (W3IWI,2976) 12/ 1/85   2:14 AM L:6
KEYS:/HOWIE/A COUPLE OF QUESTIONS ABOUT 8K CODE/

Howie, another question. Do you plan an 8k release with the latest goodies (or most of them) -- perhaps fewer multi-connects? The last 8k release you sent out (about 6 weeks ago) was pretty bullet-proof. Any problems if I make it available to locals here? The hardware handshaking and a couple of other fixes really do improve BBS operation and these guys wonder why mine works so well when theirs doesn't!

M 19111 J. Gordon Beattie Jr. (N2DSY,2990) 12/ 3/85   7:21 AM L:5
KEYS:/TNC 2 REV 1 MOD NEEDED ASAP !/
TO: (Group 95)

Has anyone published a modification for the rev 1 TNC2s so that they can support the 27256 like the rev 2 boards can ?
This is needed ASAP !
73, Gordon

M 19181 Lyle Johnson (WA7GXD,2973) 12/ 3/85 11:16 AM L:6
KEYS:/TNC 2 REV 1 MEMORY MODS/
TO: (Group 95)

gordon,
complete mods were published in the october PSR. just add the following steps. 1 - cut the trace from u24 pin 26 to wherever it goes. 2 - add a trace (jumper) from u24 pin 26 to u24 pin 28.
that's all there is toit! (see how useful PSR can be?)
lyle

M 19455 Rich Amundson (WAØJFS,2965) 12/ 3/85 11:38 PM L:19
TO: (Group 95)

TO: GROUP 95
FROM: RICH AMUNDSON
RE: X-82Ø RLI HANDSHAKING PROBLEM

TO HELP THE IOWA TRANSITION TO 145.Ø1 ON DECEMBER 1, I IMPLEMENTED THE GATEWAY FUNCTION WITH ONE PORT ON 145.Ø1 AND THE OTHER ON 147.555, BOTH AT 12ØØ BAUD TO KEEP THE STAGGLERS IN TOUCH UNTIL THEY CAN QSY.   ON DOING THIS HOWEVER, THERE RAISED ITS UGLY HEAD A HANDSHAKING PROBLEM THAT MANIFESTS ITSELF AS A FAILURE TO SEND THE ^Q AS SUCH TO TURN ON THE COMPUTER AGAIN WHEN SENDING A LONG FILE. THEN ALL THE CONNECTEE CAN DO IS LET THE BBS TIME ITSELF OUT AND LET IT RESET WHICH IT WILL DO AND EVERYTHING SEEMS OK TILL THE NEXT PERSON TRIES TO DO THE SAME OR ANAY OTHER LONG FILE. BY LONG I MEAN ANYTHING OVER ABOUT 12 OR 14 LINES OF TEXT.

THE HARDWARE IS A "STOCK X-82Ø-1" TWO TNC-1'S AND ASSOCIATED RF GEAR. ALL OF THIS GEAR WAS USED BEFORE IN THE SINGLE PORT BBS SO I AM STARTING TO SCRATCH MY HEAD.

ANY IDEAS WHERE I SHOULD LOOK TO SOLVE THIS PROBLEM? ANY SUGGESTIONS WILL BE APPRECIATED AND PROBABLY TRIED.
THANKS   WAØJFS

M 19469 Lyle Johnson (WA7GXD,2973) 12/ 4/85 12:18 AM L:19
KEYS:/TNC 2 REV 1 OWNERS/ANNOUNCEMENT/FOR GENERAL DISTRIBUTION/
TO: WA7GXD, (Group 95)

To: TNC 2 Rev 1 Owners
Re: Rev 2 Upgrades

Noting the improvements made in the TNc 2 Rev 2 design, many of you have contacted TAPR requesting information on obtaining parts for the various upgrades listed in the October, 1985 PSR Quarterly.

After listening to your calls and reading your letters, we are pleased to announce that we are ordering sufficient parts to send to every

to announce that we are ordering sufficient parts to send to every
TNC 2 REv 1 owner a "care package" consisting of the 27C256 EPROM, second 8k static RAM, RF Chokes and miscellaneous capacitors, along with a corrected copy of the PSR article.

If all goes well, the package should arrive in time to be a stocking stuffer.

The price? Hey, this is TAPR. There will be no charge...

Thank you for your support. Happy holidays from TAPR!

M 19479 Tom Clark (W3IWI,2976) 12/ 4/85 1:41 AM L:66
KEYS:/SAREX2/SHUTTLE PACKET EXPERIMENT/INFO UPDATE/
TO: (Group 95)

The following is from the W3IWI packet BBS for info:

```
W3IWI at 851204/0531: Last msg # 218, 98 active msgs
(B,D,G,H,I,J,K,L,R,S,T,U,W,X) >
Msg# TR  Size  To     From    @ BBS   Date     Title
 192 BN  2919 ALL    W3IWI           851204 SAREX2 demo
```

Many of you are aware that we are hard at work to get SAREX2 built. SAREX2 is an acronym for Shuttle Amateur Radio EXperiment; SAREX1 was the slow-scan TV that WØORE used earlier this year.

SAREX2 is to be a packet radio package consisting of a flight-hardened TAPR TNC2 coupled to a Radio Shack Model-1ØØ lap-top computer and the same Motorola "MX"-series HT used on previous amateur radio shuttle flights.

We are striving, pending approval from NASA, to have this hardware carried by Ron Parise, WA4SIR when he flies on mission 61E taking off March 6, 1986. The crew working on hardware & software to make this happen are located in Maryland, Arizona, Florida, Texas, Arizona and Iowa.

We now have prototype SAREX2 software operating on 145.Ø5 in the Balto/Wash area under the call W3IWI-5. It is under test to see if local users can make it croak! It has several unique features:

ROBOT -- This is an automatic QSO machine that will work you and assign you a unique serial number (in hex) for your QSO, and then disconnect.

WORKED beacon -- If you successfully complete a QSO, your call and serial will appear at the top of the "WORKED" beacon list. For local testing this (and other) beacons are being sent every 15 minutes, if flight they will be repeated every 15 seconds or so.

HEARD beacon -- This beacon begins with a beacon sequence number (to help piece the log together in flight) and lists the calls heard most recently.

A normal beacon (currently addressed to SAREX2) is also used to send a brief (up to 12Ø bytes) information packet.

A "meta-beacon" is also available (but is currently turned off) to send up to 1.7 kbytes (7 * 255 bytes) of information. It is planned that this feature will be used in flight to downlink a mission status bulletin every couple of minutes or so.

These features have been designed to permit a large number of packeteers to experience the thrill of space communications even if WA4SIR is not personally available.

During the mission, some time will be reserved for special packet tests including an orbital demo of store-and-forward communications (a la PACSAT); software to support this special mode is to be housed in the Model-1ØØ.

In addition to the Balto/Wash W3IWI-5 tests, WAØJFS and WØRPK are planning an aircraft demo/test over the midwest in early December.

The ROBOT supports multiple QSO's simultaneously. Balto/Wash users are invited to become abusers and to try to crash the test software. Gang up on W3IWI-5. Lambast it with simultaneous packets. Try to make it croak. Make my day!!! Meanwhile, W3IWI-5 will also continue to serve as a local coverage digipeater from its location at NASA/Goddard in Greenbelt.

Reports on how it works will be appreciated. Especially solicited are bug reports which should contain as much info as possible on what happened. It's a lot easier to fix it on the ground than when it is flying in orbit!

73, Tom

M 1948Ø Tom Clark (W3IWI,2976) 12/ 4/85   1:52 AM L:95
KEYS:/A MONTH IN THE LIFE OF THE W3IWI PBBS/THINGS ARE GROWING FAST!/
TO: (Group 95)

The following is from the W3IWI BBS. I thought some of you might like to see some statistics on how busy things can be and how fast activity is growing -- 73, Tom

==

I have just posted the November user summaries as the files SUM-NOV.DAY and SUM-NOV.USR. There were 163 different users this month (a new record) and the system was busy some 32% of the available time. This month the reliability was lower (92% availablity) than I would have liked due to a number of power transients in the middle of the day which killed things until I could get home. Another record was that a total of 1124 messages "flowed" thru the system, an average rate of 1.7 messages per hour.

Since the BBS went on in Sept. 1984 we have now handled over 1Ø,ØØØ messages. This milestone necessitated minor surgery in that WØRLI software cannot cope with 5-digit message numbers. We have now had to start message numbers over from scratch since we joined the 1Øk club.

I have noticed that late in the month we were seeing 1-3 new users each and every day. Several indicators show that activity is doubling every 4-7 months. These include the number of users, the number of messages handled, and even hassle factors! One of those indicators is the sheer volume of logging data generated by the system which is summarized in the following:

```
 Log file     Size   Ø    5Ø  1ØØ  15Ø  2ØØ  25Øk
----------   ------  |----+----|----+----|----+
LOG-SEP.84 =  11ØØ8  |*
LOG-OCT.84 =  43136  |****
LOG-NOV.84 =  8192Ø  |********
LOG-DEC.84 =  96512  |**********
LOG-JAN.85 = 114688  |***********
LOG-FEB.85 =  97Ø24  |**********
LOG-MAR.85 = 12608Ø  |*************
LOG-APR.85 = 1ØØ6Ø8  |***********
LOG-MAY.85 = 137Ø88  |**************
LOG-JUN.85 = 1152ØØ  |************
LOG-JUL.85 =  928ØØ  |*********
LOG-AUG.85 = 163584  |****************
LOG-SEP.85 = 2ØØ32Ø  |********************
LOG-OCT.85 = 21Ø816  |*********************
LOG-NOV.85 = 217984  |**********************
```

Hope you find these statistics interesting.

73, Tom

==

TO US SO QUICKLY.

SUM-NOV.DAY file:
W3IWI MailBox/GateWay Daily Usage Statistics for the Month of November, 1985

Connects vs. Time-of-Day vs. Date

Da	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	Totl
1	2	4	8	4	5	4	1	2	.	1	4	4	3	2	1	6	3	3	3	1	3	.	4	4	72
2	2	4	5	5	6	4	1	1	.	1	1	1	3	5	4	1	2	3	5	1	4	2	5	3	69
3	2	2	5	2	3	4	2	2	2	3	1	2	4	5	2	3	2	2	4	2	3	4	6	3	70
4	.	2	4	3	3	3	.	.	.	1	2	1	1	3	1	<------ off the air ----->									24
5	5	.	.	1	4	1	3	3	2	2	.	2	3	.	1	4	2	3	1	.	2	2	4	5	50
6	3	2	4	6	5	.	2	3	1	.	2	1	1	1	.	1	4	.	3	2	2	4	1	1	49
7	4	5	2	3	3	1	1	2	.	1	.	2	3	2	1	.	1	.	.	3	2	2	4	4	46
8	5	2	6	2	3	2	1	.	.	.	2	1	4	1	4	1	2	1	1	2	1	.	2	4	47
9	7	4	3	2	4	4	.	.	.	.	4	.	4	4	1	1	1	1	3	4	2	5	3	5	62
10	1	1	5	4	2	2	3	1	.	.	1	2	<----------- off the air ---------->												22
11	<---------------- off the air ----------->														3	1	.	3	3	2	<--- off -->				12
12	<--- off -->				1	2	2	.	.	.	1	3	2	2	1	1	2	1	2	3	2	1	3	8	37
13	6	2	12	4	3	1	3	2	1	2	.	3	2	.	.	4	2	5	<---- off ------->						52
14	<- off ->			3	6	4	1	2	.	2	2	4	4	.	2	1	1	2	.	1	2	9	2	.	48
15	13	6	3	4	6	3	2	1	.	1	1	1	.	1	1	2	5	5	2	3	4	1	4	5	74
16	6	4	3	9	7	3	.	2	.	.	1	3	5	4	1	4	3	4	2	1	2	3	5	3	75
17	2	5	4	1	4	6	7	2	2	1	1	1	2	6	4	2	4	3	3	7	2	5	5	4	83
18	3	5	4	5	2	4	4	1	1	1	1	4	1	3	2	2	4	1	1	<-- off --->				2	51
19	6	6	4	6	6	1	2	.	2	1	4	7	1	.	1	2	1	2	3	1	.	2	6	3	67
20	3	5	4	3	3	5	.	1	.	.	1	5	2	.	1	1	2	7	1	.	3	3	2	4	56
21	.	7	6	4	4	3	2	1	1	.	.	4	3	4	1	.	5	<----- off --------->							45
22	4	3	4	9	4	3	2	.	1	1	.	4	5	.	1	1	2	2	3	12	5	3	6	3	78
23	3	4	4	5	10	.	.	2	.	1	1	1	1	.	2	2	3	2	.	2	3	4	6	3	59
24	3	5	6	4	5	7	1	2	1	.	.	.	1	3	.	3	2	3	4	3	6	5	3	3	70
25	1	5	2	7	4	6	1	2	1	.	1	1	1	2	3	3	2	3	2	5	4	4	4	5	69
26	7	6	6	4	2	3	1	2	1	1	.	3	3	1	2	1	1	1	2	4	1	2	2	2	58
27	2	3	5	5	3	3	5	2	1	3	3	3	3	.	3	3	2	3	4	2	3	5	4	4	74
28	5	3	5	3	1	1	3	3	2	.	1	3	3	4	4	4	4	4	3	5	5	2	5	2	75
29	2	5	4	3	2	5	3	1	.	.	1	2	7	2	6	3	<---- off ---->					3	8	3	60
30	3	1	7	4	3	5	1	2	2	.	1	1	1	.	.	.	2	6	5	2	5	4	3	9	67
	100	101	125	115	114	90	54	42	21	23	37	69	73	55	53	57	64	70	60	68	66	75	97	92	1721

Total Time Available : Approx 652 hours = 91.4% reliability
Total System Connect Time: 12487 minutes (208 hours,7 minutes)= 31.9% of avail.
Average Time per Connect : 7.255665 minutes
Average User Connect Time: 76.60736 minutes
Total number of users : 163
Average connects per user: 10.56

==

I won't bore you with the individual user statistics in the file SUM-NOV.USR. Hope you found the material interesting. 73, Tom

M 20849 Rich Amundson (WA0JFS,2965) 12/ 7/85 10:04 PM L:15
KEYS:/SAREX2 TESTING/
TO: (Group 95)

TO: HOWIE, TOM CLARK, LYLE

FROM: RICH AMUNDSON

RE:SAREX2 TEST

W0RPK AND I HAD THE BETA TEST SOFTWARE ON THE AIR FROM THIS QTH FOR AN HOUR TONIGHT AND HAD OVER 400 CONNECTS. NOONE WAS ABLE TO MAKE IT BURP OR EVER BREATH HARD. ONLY OCCASIONALLY DID IT MAKE IT INTO THE FOURTH LEVEL. IT LOOKS GOOD FROM THIS END HOWIE AND I DID GET RAPLH THE INFO.

NOW, IF ONLY THE WEATHER WILL COOPERATE WEDNESDAY NIGHT, WE SHOULD BE ABLE TO GIVE IT A STIFFER TEST.

73'S AAND THANKS TO YOU ALL FOR GETTING THE INFO AND EPROMS TO US SO QUICKLY.

RICH WAØJFS

M 21ØØ3 Skip Hansen (WB6YMH,2964) 12/ 8/85 1:52 PM L:15
KEYS:/NNC SCSI HARD DISK CONTROLLER/
TO: (Group 95)

To: Potential NNC software developers
From: Skip WB6YMH
Subject: Mass storage for NNC software development.

My company has a limited number of SCSI bus hard disk controllers available for ST-5Ø6 style hard disk drives. The controllers are DTC-51ØB's which are surplus to our needs, but are new. The price is $6Ø. There are several companies offering good deals on 1Ø Mb hard disk drives which would allow a 1Ø Mb hard disk to be interfaced to the NNC for around the cost of two floppies. I have written a CP/M 2.2 BIOS which supports a Segate ST-412 with this controller and would be happy to make available. If anyone is interested let me know. If nothing else it's a hell of a good way to make sure the SCSI port gets a good thrashing!

73's Skip WB6YMH

M 2112Ø Tom Clark (W3IWI,2976) 12/ 9/85 12:58 AM L:42
KEYS:/SAREX2 NEWS/FROM NORTHERN CALIFORNIA AND IOWA/
TO: (Group 95)

SAREX2 news

A brief note to tell you that SAREX2 software is now operating in California. I am in San Francisco on a business trip and brought my portable SAREX2 demo station with me. It has now been on the air for a bit over a day and has logged 8Ø QSO's with some 25 different stations on 145.Ø9 operating thru the WD6CMU-1 digipeater.

73, Tom

Posted: Sun Dec 8, 1985 6:Ø1 PM GMT Msg: PGIF-2329-2579
From: RWALLIO
To: tclark
CC: pacsat
SJSAREX-2ATESTINGeiniIOWAn Iowa

PROMs burned with Howie's SAREX-2 software arrived from TAPR and W3IWI on Friday last. One copy was immediately installed at WØRPK to get ready for a coordinated test which occurred this past Saturday evening. The other copy was installed at WA2GTM preparing for the aeronautical operation scheduled for this coming Wednesday.
The test on Saturday evening resulted in slightly over 4ØØ-QSOs in one-hour with about 1Ø-stations participating. In slightly over an hour, the total QSO count was 425.
When I arrived back home from the WAØJFS test site, I found that the disk file I had openned had collected 9ØK-bytes of SAREX-2 beacons. This file, and what we collect on Wednesday, will be used for beacon processing program test data.
Plans for our aeronautical operation are complete and we are watching the weather closely. A last minute briefing will occur on the Tuesday evening AMSAT nets.
Ralph Wallio, WØRPK for CITS

[W3IWI note -- I apologize for the garble at the start of the Iowa report. This msg is being sent using a Model-1ØØ which seemed to have gotten its handshaking fouled up. -- Tom]

C2973 CC47 Jack Brindle (WA4FIB,2963) 12/ 5/85   1:17 AM L:57

Phil; Yes, I am here, listening, and very interested. How do I get a copy? I will send an SASE REAL SOON NOW, as soon as I can crawl out from under my MacPacket/TNC2term development. Please give me your address again, I managed to lose it in the move up to Atlanta.

To those into the argument of space taken in systems, I have calculated the amount of space required for tables in a PVC type system. These same calculation could probably be extended to other systems as well (VCs & datagrams)

The X.25 network layer format allows a 12 bit Logical Channel Identifier (LCI). Using two of these bits for encoding packet priority, we then have a ten bit LCI. This allows for 1Ø24 logical channels. This also means that in a PVC system we can have 1Ø24 nodes (4Ø96 if we use some other means to convey priority). Each node requires an entry in the node's routing table, consisting of the layer 2 channel to send the packet on to forward it to its destination. Thus each entry consists of one byte. There might be a second byte for alternate routing, or an alternate LCI could be used instead. This means that the outbound routing table would require a maximum of 1Ø24 bytes of space. Inbound routing tables are also required for every node the present node talks to at the link layer. These tables consist of a single entry for the required LCI to forward the packet. The table is indexed by the inbound LCI. Thhus the table would contain a maximum of 1Ø24 entries each of which would be 1Ø bits in length. Since we are using byte machines, we shall assume that the entries actually use 16 bits even though the upper 6 bits are zeros. This means that we now need a table of 1Ø24*2 bytes (2Ø48 bytes) maximum for each node the current node talks with AT THE LINK LAYER. Being practical, we should only be able to connect directly to 2 or three other nodes. Our table requirements then would be 5 kilobytes for 2 directly connected nodes, or 7 kb for 3. In light of the current price of 8Kx8 ram chips, this will allow NNC ram to be used for message buffering. We now begin to see that we really don't need a lot of RAM in our NNCs for routing tables, and that the amount of ram is governed instead by the number of packets buffered before sending them on to their destination. This will not be much in our present AX.25 link layer protocol, but as we begin to go to higher data rates, we may wish to extend the possible maxframe values so that 127 packets may be sent in a group. This, indeed, is where our RAM space will be used.

Oh yes - those thinking that 1Ø24 nodes is not very many in a system, it is when one begins to see packet as taking a regional approach, the way the ARRL traffic network does. I believe we will see regional networks that are bridged though system bridges and gateways. This will make the tasks of system upkeep and maintenance bearable, and keep network control "local".

One last thought. For those that believe that any hodgepodge of a system will do since "this is only ham radio", I urge you to take a look at what is happening in the radio packet networking commercial community. Several companies are pushing packet products for commercial use based on the AX.25 protocol (e.g. AEA, Kantronics and GLB, to name a few). This is not necessarily bad, but whatever protocols we decide on will probably get pushed onto the commercial users. Since each networking protocol has its good points and bad, and each is better used in specific systems, an end user may get stuck with a totally unsatisfactory system. It will probably be quite some time before any of the commercial folks realize this, but I bet it will upset a few. For example, a small company whose network is fixed doesn't need the versatility of a VC system (but a PVC system would work exceedingly well). I don't suggest that we design our systems based on the needs of others, but I do believe that the commercial folks better watch what they are pushing on the unsuspecting commercial world, lest they be overrun by a superior networking system. Remember, this is ham radio, and the commercial folks better remember that we do this for the enjoyment. Right Phil?

C2974 CC1Ø6 David Cheek (WA5MWD,2966) 12/ 5/85   1:Ø9 AM L:248
KEYS:/TEXNET/NETWORK5/

An Introduction to Networks
part 5

by T.C. McDermott, N5EG
networks SIG, TPRS

Before we get far into the discussion of network software requirements, I would like to make a correction to one of the previous articles. We have chosen AX.25 as the link-layer protocol internal to the network, with the exception that the maximum data size is 512 bytes. Since the user of TEXNET may generate a packet up to 256 bytes long, and since the network will overlay it's level-3 protocol onto that packet, then internal to the network packets larger than 256 bytes can exist. An alternative to this is to fragment packets, and it was decided that this was an unnecessary complication at this time.

In a previous article we saw that algorithms that assume slightly unreliable radio paths can be chosen to minimize the degradation in throughput suffered by packets (the HOP-TO-HOP algorithm, for example).

We have also seen that all the hardware that is really necessary to build a node includes 2 radios, 2 modems, and the node control processor (plus minor items like: sites, towers, feedlines, power, people to maintain hardware, money, time, etc.).

What then has delayed the introduction of networks to the amateur community? Simply, it the great level of complication in the software that is necessary to build a network. We will see that a network requires two layers of protocol, not one layer, as we use in the AX.25 link layer. What are the two layers of information, and what purpose do they serve?

Let's define some terms here, since we will use them frequently in the following discussion.

- LAN : Local Area Network. This is the part of the network that the users are connected to the nodes through. usually on 2-meters, at 1200 BPS.
- IP : InterProcessor. The part of the network that the nodes talk to each other on. usually on 220 Mhz., at 9600 BPS.
- source_user : this is the Ham, using a TNC, that generates the original information to be transmitted. This Ham may have typed it in on a CRT, or may be sending a disk file, or it may be a BBS sending a message. It is the point where ASCII text gets translated to HDLC.
- entry_node : This is the entry point to the network. When a HAM wants to use the network, this HAM will CONNECT to this node, via the normal method for the TNC to connect to anything.
- exit_node : This is the exit point from the network. It is the place where the node is close to the desired ultimate consumer of the information. The information leaves the network, via AX.25 at this point.
- dest_user : This is the destination user, the consumer of the information that is being transmitted. Just as source_user is connected to entry_node, dest_user is connected to exit_node via a AX.25 link-level connection.

When you use a TNC to connect to someone, the AX.25 packet contains two key pieces of information, the source, and the destination. There is never any confusion here. Source and Destination change whenever the direction of transmission changes. Also, since this is a link-layer protocol, there is never any confusion at to where destination might be located. If it is not within range of the radio, then no connection is ever established!

What is different about a network? For one thing, the network doesn't really "know" where the destination is. It must

network doesn't really know where the destination is. It must ask the source_user where to find dest_user (or it might look up dest_user in a table, but this gets complicated since users tend to move around a lot). So the network needs to know where to find dest_user, and the answer is supplied as "dest_user is located NEAR to a particular network node, called exit_node". Now, it would be nice for dest_user to be able to send information back to source_user, who happens to be located at entry_node. Thus when setting up a connection, we see that the network, prior to the exchange of user data, must establish where the users are located, so that it can send that data to the right place.

Several divergent opinions of how to get that user data to that place could now burden this discussion. Suffice it to say that we have chosen a particular method for TEXNET for what we perceive to have a simple implementation method.

Our method is as follows:

1. Entry_node receives a packet from source_user.
2. The network uses the AX.25 fields to tell it who source_user is, and then strips off the AX.25 header.
3. It now adds a brand-new header, called TEXNET IP3. (InterProcessor layer-3). This header contains the length of the data part of the packet, exit_node, dest_user, entry_node, and a one byte control field, followed by the packet.
4. The exit_node field is examined by a routine called ROUTE, which figures out which is the next node of the network that should get the message. This next node will obviously be one that is closer to exit_node than this node is.
5. An AX.25 header is now built onto the front of this big packet. As the source field in this header, it will contain our node name. As the destination, it will contain the name of the further-along node (that is adjacent to us) that was supplied by ROUTE.
6. This complete unit will now be transmitted at 9600 baud, using all of the rules of AX.25, to the next node.
7. The next node will strip off the AX.25 header after it receives the packet. It will examine exit_node to see if exit_node = our_node_name. If it does match, then this node is the exit_node, goto 8, else we are not the exit_node, and the packet must be further propogated: Goto 4 to continue the propogation of the packet down the network.
8. When exit_node is reached, the field dest_user in the IP3 header is examined, and the correct information found for that packet. The IP3 header is stripped off, and the correct layer 2 AX.25 information, with this node name being the source, and dest_user being the destination is formed. This packet is then sent using AX.25 methods to the destination user.

Although this sounds a little complicated, its really not. INSIDE the network, any packet starts with a layer-2 header specifing which are the immediately-adjacent nodes that are exchanging the packet. Next in the packet is the IP3 header, which contain the endpoints of the connection. Finally is the actual data itself. Note that we can place several different IP3-DATA sets within a single layer-2 envelope, as long as the different packets are going the same dirrection. Thus we have MULTIPLEXED packets from different users into a single layer-2 unit.

This method has some advantages. One is that HOP-TO-HOP acknowledge is explicitly a function of layer-2, since each node must acknowledge the receipt from the previous node by the AX.25 method. Secondly, an interviening node need only examine the exit_node field to decide if it needs to process the packet, otherwise the IP3 portion remains UNMODIFIED by that node. Layer 2 source and destination will be changed since the node that received the packet becomes the new source, and ROUTE will now supply a destination node closer to the exit_node.

The complete layer-3 protocol includes a few more details.

such as a software state diagram to describe the exact method that the network uses to do everything, such as set-up and to tear-down a circuit. It also contains some instructions for handling error cases. The layer-3 machine does not worry about the reliability of the radio paths, that is the responsibility of layer-2. It does however check for inconsistent activities, and erroneous values in the control field of the IP3.

What does the control field contain? It tells us whether the information that follows is user_data (which it is most of the time), or whether the information that follows is supervisory information, for use ONLY within the network. The possible supervisory commands include these:

Circuit establishment request
Circuit establishment acknowledgement
Circuit disconnect
Circuit disconnect acknowlegement / or estab. failure
Error - Network transmission failure
Error - User transmission failure
Traffic statistics request
Traffic statistics response
Congestion control - flow on
Congestion control - flow off
Special "fire code" sequence request
Processor reset acknowledge

The circuit supervisory commands allow the two endpoints of the network to exchange information about setting up or taking down a connection. When setting up a connection, the entry_node can tell the exit_node who source_user, entry_node, and dest_user are, and any digipeaters that exit_node may have to go through to get to dest_user. Exit_node will either be sucessful at setting up the connection to dest_user, or it will not. It sends back this result to entry_node, so that source_user can be notified.

One of the previous articles described the special hardware state-machine on each NCP card that will detect a long sequence of characters that obey HDLC coding rules and directly generate a reset pulse to the microprocessor. The "fire code" command is the way that the network makes sure that the request gets to the node that needs to be reset, and that the node PRIOR to the target node will transmit the special sequence. Each node has a different state ROM, and thus a different sequence to reset it. Each node "knows" the code for it's neighbors.

Congestion control can be a very complicated subject. Congestion control and flow control ARE NOT THE SAME THING. As an illustration of this point, assume that source_user is generating packets quickly, and injecting them into entry_node. Entry_node routes them down the network to exit_node. Exit_node, however is having a difficult time trying to get the packets to dest_user. They get through, but there are a lot of collisions, and the leave exit_node slowly. Eventually, there will be a network congestion problem. User data will start filling up all of the available RAM at the intermediate nodes. Finally, the packets will back up through the network until entry_node no longer has any buffers to take packets from source_user. So entry_node will use the AX.25 RNR packet to tell source_user to flow-off. TOO LATE, the network is already hopelessly congested. What should have happened is that exit_node noticed that packets were coming in for dest_user faster than they were being delivered. When exit_node notices the problem, it sends a message to entry_node telling source_user to stop (entry_node uses the AX.25 RNR). Now the network is not congested, and the source_user is stopped. Exit_node must tell entry_node to re-start when conditions will allow. So entry_node has TWO conditions to check for: 1. Flow-off the source_user if entry_node is low on buffers, or 2. Flow-off the source_user if exit_node cannot deliver packets. Only when BOTH conditions have cleared can source_user inject more packets into the network.

There are many other methods of congestion control, and this is obviously a simple one with some defficiencies, but it is easy to implement.

One of the difficult points to bring up is lack of a layer 3 protocol between the user and the node. Since only the link-layer protocol is currently defined in AX.25, certain problems arise in the operation of the network.

When source_user first connects to the network, he must engage an interactive question-and-answer session so that the network knows to whom source_user wants to connect. A layer 3 protocol would provide this facility aunection establishment.

When exit_node connects to dest_user, the destination user thinks it is connected to the network, the dest_user does not know the name of source_user unless the network specifically tells dest_user PRIOR to delivering traffic to dest_user. Again, this facility would be provided by a layer 3 protocol in AX.25.

Although this doesn't seem too major a point, consider the operation of BBS's with a network. You connect through a network to a BBS. What does the BBS think your callsign is? The callsign of the network, of course. How does a BBS perform mail-forwarding through a network? Each BBS could "kludge" a method to use to the network, but a standard method that works REGARDLESS of who designed, built, and programmed the network would really be nice. This should be in the domain of the AX.25 layer 3 protocol.

One of the objectives in the protocol we desinged for the layer-3 inside of the network (IP) was to allow the operation of a device known as a Network Bulletin Board (NBBS). Since the TEXNET IP3 header contains enough information, a NBBS can be a device that talks to the network AT 9600 baud directly on the IP link. It must emulate a NCP, but it has the information to do that. Ideally, it would be a multi-user NBBS, since the IP supports multiple users per NCP.

For those who have read this far, this network can be described as the type that presents a virtual-circuit interface to the host, but uses a datagram method inside the subnetwork. This makes the network similar to ARPANET, and not like SNA, or DECNET. (See Tannenbaum, Fig. 5-4)[1].

Part 6 of this series will describe one way to put together a software package to perform these functions. Currently, we have psuedo-coded 30 software modules for TEXNET. Certain rules and programming metholodogy can help to ease the construction difficulties with a complex problem.

----

[1] I forgot to credit the fundamental work of the subject in one of my prior articles.
"Computer Networks" by Andrew S. Tannenbaum; Prentice-Hall, Inc. C 1981.

C2974 CC107 Phil R. Karn (ka9q,2979) 12/ 6/85 7:27 PM L:61
KEYS:/TEXNET COMMENTS/

A few comments on Part 5 of the TEXNET articles that was just posted.

It is interesting to watch as various members of the amateur community slowly re-invent each aspect of the ARPA Internet protocols, step by step, even if they do not realize that this is what they are doing. With a few differences I'll talk about later, this is what TEXNET seems to be doing.

The use of encapsulation to send a network datagram across a series of links is fundamental to how the Internet operates. TEXNET seems to be doing the same thing, but it assumes that the only link level protocol that will ever be used is AX.25. Why? There are lots of link level protocols out there, and if you can encapsulate your datagrams in all of them transparently, you've created a much more flexible

in all of them transparently, you've created a much more flexible
way of doing things. The link protocol can be customized for the particular situation (i.e., providing hop-by-hop acks if they are deemed necessary), while the higher level network protocol can worry about things like global addressing and routing. I believe that the two need to be kept completely separate, with the network level datagram service assuming as little as possible about the underlying link level protocol.

The article points out the fact that user interfacing with the network is clumsy, because you need some way to tell the network what to do with your packets. This is very true! Eventually it should become obvious that the way to solve this problem is to simplify the user/network interface as much as possible, i.e., by the provision of a simple datagram interface. This also means that the end-to-end protocol or "transport" protocol, for which we now use AX.25 LAPB even though we continue to call it a "link" protocol will have to reside in the user's TNC. I believe that the only practical way to do this is by adding the full-blown network transport protocol (i.e., TCP) to each user's TNC. I have shown with the code I've recently written that this can be done in about 6K bytes, which is probably smaller than just the command processor that would be required by a connection-oriented network interface.

The later comments about the difference between congestion control and flow control are very true. I've been trying to get this point across for some time. One thing that should be apparent from the discussion is that "backpressure" flow control schemes are to be avoided whenever possible, since they allow a user to leave packets unreceived in the network for indefinite periods of time. If a user isn't ready to receive a packet, it shouldn't be sent in the first place. All end-to-end transport protocols, such as TCP and AX.25 as used presently, have flow control windows to provide this. Link level flow control, if used at all, should be used only to provide hop-by-hop control over traffic, not as a substitute for end-to-end flow control. A well-designed transport protocol will also attempt to generate as few packets as possible to carry a given amount of data, and this is something that modern versions of TCP (with the "Nagle Rule") do very well. They also react to special messages from the network that indicate congestion by backing off at the rate at which they inject traffic into the network.

In summary, then, I would argue strongly that the TEXNET design team take a close look at the Internet protocols, since their overall design aims and approaches appear to be compatible with those of TEXNET. I would be happy to provide any documentation if required.

73, Phil

C2974 CC108 Howard Goldstein (N2WX,2987) 12/ 8/85 2:30 PM L:75
KEYS:/I THINK I CAN I KNOW I CAN I THINK I CAN/
A: 107

Finally there is just so much BULLSHIT and SLANDER that its time to correct some of these statements!

Its a crock of shit that "6K bytes is probably smaller than just the command processor that would be required by a connection oriented network interface." I don't know what kind of programming techniques prevail in the TCP/IP world but if it takes 6K bytes, someone needs to go back to school. Is anyone else tired of pontifications from the ivory tower - in the real world, not this high flake intelluctal crap that we've got reams of paper wasted over now, things are not as clearcut as you've alledged them to be - damn they aren't even LIKE the way you've alledged them. A VC implementation takes around 6K bytes for everything after ax2512v2 - X.25 13, X.75, the command interface, the whole kit and kaboodle.

... difference in expectations has direct effect on the capabilities of

But are we here to argue about coding efficiency? Nooooo, not if memory is so cheap

No matter how you cut up the datagramme approach we're gonna wind up with a HIGH FLAKE NETWORK OF DIGIPEATERS.

{In case no one's haven't noticed, there is already a 'standard' link level protocol. Sure there are lots of others - so how in hell did the STUPID and INCONSISTENT "UI" handling manage to get in to pervert this otherwise pristine standard? Has anyone ever tried to reconcile the imbecilic way UI frames are supposed to be handled in L2V2 with the rest of the document? Kind of makes one wonder how standards get formulated. This case is a perfect example of the "klugeyness" inherent in the datagramme approach. What a mess, a new piece added whenever someone gets a new idea they like. No two people running the same thing. More later

}

Transports- Very nice, lets put a transport protocol in each switch and then we won't need a link protocol. Nifty, but it's still digis, plus everyone has to have some complex and redundant transport protocol in their TNC. Can we see some code running in a TNC? AX.25 LEVEL 3 EXISTS - TODAY - AND RUNS IN A TNC OVER AX.25 LEVEL 2.

Everyone needs to remember that the TCP/IP bunch wants us to use a connectionless network. Can you all imagine trying to CONNECT to someone VIA 2Ø or 3Ø digipeaters? That's the approach being touted, friends, its end-to-end all the way - WHETHER YOUR SESSION REQUIRES END-END ACKS OR NOT you're gonna have to use them.

I think this should be a LOCAL choice a user could specify if he wants it.

Yes, congestion is quite different from flow control, and if a user isn't ready to receive a packet the question is not leaving packets unreceived in the network (again, memory is supposed to be cheap - right????) but how long it takes for a sender to realize and get data through again after he's turned back on. Takes a while for the "clogged" end to turn the other side back on with TCP/IP, no? Yes? Flow control is an exception ax.25 l3 plans for. It's designed to make buffer management VERY EASY no matter how many VCs go across an interface (okay, mod256 might be desirable for bunches of them) and optimizes throughput for everyone concerned.

In fact, I personally found it much easier to install AX.25 level 3 and its practically identical IP cohort X.75 than level 2 was! Someone working with a high level language should be able breeze through the Network layer in a week. Terry's proposal was definitely ahead its time. I think its time has now arrived.

On congestion control: How does a stateless switch deal with priorities and congestion? Correct me if I'm wrong, I'm under the impression that it TRASHES stuff it can't deal with without any indication. Just like what we're doing now with digis. May we have an example of how a stateless switch acts when its plate is full?

In summary, there seems to be a fundamental irreconcilable difference between the datagramme and VC approaches to network. I contend this difference has to do with the expectations each community has of it's media.

Datagrammites expect an infinite # of links to be error prone and rarely available (perhaps as a result of nuclear detonations), in a way similar to the way one would try to have a QSO in UNPROTO mode.

A virtual circuit network expects a finite number of error free links to be available. Links like that which are already provided by AX.25 L2V2.

This difference in expectations has direct effect on the capabilities of

This difference in expectations has direct effect on the capabilities of the network.

Planning in a datagramme world is like doing 5mph under speed limit on a coast-coast drive the whole way. No one does it, despite the better chances of living through the trip and not getting a ticket.

And those folks with the $$ fly their own learjets exactly where they have to go without concerning themselves with drunk drivers and local yokels.

I do not expect the amateur community to be able to afford learjets (or high speed leased lines between switches), nor do I think there's a use for a network convinced every transmission was puked on, corrupted, and otherwise violated.

73 Howie

C2987 CC244 Howard Goldstein (N2WX,2987) 12/ 1/85 3:59 PM L:10
KEYS:/8K RELEASE/
A: 243

Tom, I really would rather not have a seperate 8K release just to keep things easy to maintain.

However there is probably a need to support the unmodified boards around so there will be one more release to kind of tie up the loose ends where there isn't a need/desire to upgrade to rev2.

If they can't wait a little while longer go ahead and let them have the release. But they're on their own since there's known faults in in that last 8K.......

73 Howie

C2987 CC245 Howard Goldstein (N2WX,2987) 12/ 3/85 10:34 AM L:73
(ORIG.) 12/ 3/85 1:23 AM L:16
KEYS:/1.1.2 AVAILABLE/EXTENSIVE SOFTWARE RELEASE NOTES/+READ THIS ITEM FOR .HEX/

To: Beta
Fm: Howie N2WX
Re: 1.1.2 notes
Dist: Beta

Please +READ C2987 CC245 for the .HEX of the release described herein.

1.1.2: Changes in this release are virtually all attributable to feedback from Beta testers. Therefore, if you like anything in it, pat yourselves on the back. This is as much your release as it is anyone's!

Time to drop some names. Especially thank the following individuals (in no particular order) for their input is represented by the features and fixes of this release: KV7B & D, WA7GXD, W3IWI, WB6YMH, AD7I, and KY2D. (Is your call missing from this list? How about downloading a release and making some comments?!)

Here are notes releated to 1.1.2:

FIXES occupy much of these notes. The following has been repaired, I think:

o- Flow release causing data loss of xmit data

o- Anamolous disconnect when in S16 link state

o- Retry-out handling when version 2 of 12 works like table now

o- ID doubling fixed (i think -- could not dupe the bug here)

o- Sluggish carrier detect repaired, the TNC is much friendlier to other stns now

o- Monitor headers aren't displayed for received data from a link when MCON ON and MON ON

o- NULLS are handled in int routines, all combinations supported w/o loss

o- CTEXT packetized at connect time instead of traveling through the queues

o- NULL artifact from true-break transparent escape is now captured

o- rare corruption of data with "cmd:" repaired

o- 1Ø second timers (BEACON, CHECK) work at 1Ø sec intervals now.

o- and finally, a very troubling intermittent that caused innumerable problems like more loss of link state, strange stuff in cmd parameters etc.

CHANGES not result of bugs:

o- ^Y handled much faster (inside async int routine)

o- UA responses while in link setup mode are IGNORED if they differ from the attempted path

o- Number-only calls allowed to support links to AX.121NA switches...

o- The date of assembly (mm/dd/yy) appended to the release number

    ex: Release 1.1.2 12/Ø3/85 - 16K RAM

o- Interrupt handling speeded up at least two times. Support for 96ØØ both sides, fulldup is gauranteed with standard (2.45Mhz clock). Support for 19.2Kbaud also assured with appropriate hardware mods for 4.9Mhz and faster I/O clocks.

(My tests used two TNC2's, both running 96ØØ sync and async, FULLDUP ON MAX 7 PACL Ø in trans and CONV mode (ECHO ON and OFF) could not drop characters thru a link of six fulldup digipeaters and a 28K file.)

o- The overflow buffer expanded to 256 characters (overflow after XOFF sent)

NEW FEATURES:

o- Failure of a CONPERMed link results in a

"*** LINK OUT OF ORDER, possible data loss [opt. daytime stamp]" message

o- the command   LCStream ON|OFF   [default ON] added.

When LCStream is ON, the character immediately following the streamswitch character is converted to upper case before being acted upon. When off, the case is significant.

o- the command   NOmode ON|OFF   [defualt OFF] added.

When the parameter is ON, the TNC will -NEVER- jump on its own between cmd: and conv or trans, or vice versa. Only user commands (CONV, TRANS, or ^C) may change the typein mode.

If the NOMODE parm is OFF then mode jumping is handled according to NEWMODE and the rev 1 TNC 2 manual.

That's it for 1.1.2. As usual, all comments are appreciated, and most receive immediate attention.

Thank you everyone for your invaluable, visible (and I hope, continuing) input to this project. 73 Howie

PS: releases prior to 1.1.2 are NOT gauranteed for 9600 baud, and should not be relied upon in any mode at this data rate!

M 21226 GARY GARRIOTT (VITA,568) 12/ 9/85 10:18 AM L:52
KEYS:/ON TO ETHIOPIA!/
TO: N2DSY, W2VY, (Group 95)

THOMAS AND GORDON--THIS IS THE STOCK STATEMENT THAT HAS GONE OUT TO OTHERS THAT PREVIOUSLY EXPRESSED INTEREST....SEND ME YOUR QTH IF STILL INTERESTED SO I CAN SEND YOU PROJECT DESCRIPTION...THANX, GARY

9 December 1985

Dear OM:

Some months ago you responded to an inquiry regarding a possible demonstration project in Ethiopia using packet radio. I am pleased to announce that preliminary approval has been acquired from the Ethiopian government to conduct the project.

Attached is a brief project description. Three TAPR TNC2's are being acquired along with Radio Shack Model 200 computers to be used in the demonstration sites. These units will be "retrofitted" on existing Motorola HF SSB transceivers.

Given the nature of the project it is very important to have tested the system in advance of actually arrriving in Ethiopia. This is because the system should be viewed as fully functional with as little experimentation as necessary while actually in the country (government officials will no doubt be looking over the consultants' shoulders the entire time). this probably means about a week's worth of testing here in the us.

Some preliminary training of both American and Ethiopian personnel will also be conducted during the three weeks or so that the two consultants will be in the country.

Please be advised that this is a volunteer activity, although all expenses (airfare, meals, lodging) will be paid through VITA. We would like to do the project as soon after the holiday season as possible.

If you are still interested, please contact me as soon as possible. If you have a recent resume or curriculum vitae, forward that as well.

Needless to say, the use of packet radio technology in disaster/relief communications is exciting and could solve a host of problems associated with existing HF radio. To my knowledge, this will be the first demo of packet anywhere in the Third World and will generate a huge amount of interest if successful.

73,

Gary Garriott (WA9FMQ)
Manager
Information Technology

M 21392 Rich Amundson (WA0JFS,2965) 12/ 9/85 10:47 PM L:25
KEYS:/SAREX? TESTS/

KEYS:/SAREX2 TESTS/
TO: (Group 95)

TO: GROUP 95(PARTICULARLY MIDWEST)

FROM: RICH AMUNDSON

RE: SAREX2 TESTING

The flight test of the SAREX2 hardware hopefully taking place Wednesday December 11 will be using the call of NØBKB-1 (listen for beacons for last minute changes) and will have at least 25 watts of rf power available to partially make for the lack of a 1ØØ mile altitude. We will be about 1Ø,ØØØ feet msl around the Greenfield, IA airport which has a location of Latitude 41-19-36N and a Longitude of 94-26-52W. We will be going in shallow circles to try and prevent shadowing of the antenna by the aircraft. WØRPK will be conducting a coordination net for anyone with questions on the AMSET 8Ø Meter frequency during the flight and will be in contact with the aircraft.

We invite all who can hear the beacons to try and connect with us to see if the system can be (ab)used. Our downlink frequency will be 145.55 and uplink _6ØØ or the standard repeater split of 144.95.

Hope to hear many of you on packet Wednesday from about 193Ø CST until
.

I HOPE THIS CAME OUT AL RIGHT. Telenet did a job on it. 73's
Rich Amundson WAØJFS

M 21432 Phil R. Karn (ka9q,2979) 12/1Ø/85 2:34 AM L:1Ø
KEYS:/NONTRANSPARENT MAILERS/
TO: (Group 95)

To all DRNET users:

Please avoid generating lines that consist of a single period at the beginning of a line. Several other electronic mail systems (e.g., Telenet Telemail and ARPA Internet mail) treat such lines as "end of message" with annoying results when you're trying to relay messages.

Thanks,
Phil

M 22Ø3Ø GARY GARRIOTT (VITA,568) 12/11/85 2:25 PM L:6
KEYS:/VOLS FOR TNC 2 CONSTRUCTION/
TO: (Group 95)

WE ARE LOOKING FOR VOLUNTEERS WHO MIGHT BE INTERESTED IN PUTTING TOGETHER THREE (3) TAPR TNC 2 KITS FOR THE ETHIOPIA PACKET DEMO PROJECT. ANY TAKERS? CAN YOU THINK OF A BETTER WAY TO SPEND YOUR CHRISTMAS BREAK??

JUST LEAVE A MSG IF INTERESTED....THANX AND 73, GARY

M 22Ø65 Pete Eaton (WB9FLW,297Ø) 12/11/85 3:48 PM L:8
KEYS:/HAZELTURKEY FIXED..BACK ONLINE/
TO: (Group 95)

Hi Gang!

Just a note to let you know I got the Hazelturkey terminal working again, and am online once again.

Did I miss any good fight}?

Pete

M 22219 Rich Amundson (WAØJFS,2965) 12/12/85 12:22 AM L:18
TO: (Group 95)

TO: GROUP 95
FROM: RICH AMUNDSON
RE: SAREX2TESTING

WELL, THE FLIGHT WENT FAIRLY WELL HERE TONIGHT IN SPITE OF SOME GLITCHES WITH THE AIRPLANE. WE HAD TO CUT THE FLIGHT SHORT BUT IN ONE HOUR OF OPERATION WE HAD 515 CONNECTS AND WE DID SEE ON THE ONBOARD M1ØØ THAT UP TO 8 CHANNELS WERE ACTIVE SIMULTANEOUSLY. MORE DETAILED INFO WILL BE AVAILABLE WHEN WØRPK HAS A CHANCE TO PERUSE THE DISK FILE CLOSER. IT DOES ACCUMULATE DATA AS HE TOLD US HE HAD 8ØK WITHIN THE FIRST HALF HOUR.

HOWIE, WE FOUND NO PROBLEMS AT ALL WITH THE SOFTWARE FROM THE OPERATORS POINT OF VIEW AND MORE WILL BE TOLD ON EXAMINATION OF THE FILES.

TOM, LARRY, NØBKB, HAD A REPORTER OUT AT THE AIRPORT BEFORE WE LEFT (FROM THE LOCAL NEWSPAPER) WHO TOOK SOME PICTURES SO WE WILL FORWARD SOME AND THE NEWS COPY AS SOON AS IT IS AVAILABLE.
73,S AND FILM AT 1Ø RICH WAØJFS

C2973 CC48 Thomas A. Moulton (W2VY,995) 12/1Ø/85 11:34 AM L:21
KEYS:/A DIFFERENT VC BETWEEN EACH NODE IN A CONNECTION FROM USER TO USER/
A: 47

Jack,
I think you misunderstand where X.25 VC's start and end.

In a X.25 based network (commerical or not) an X.25 link is made between two nodes (stations) and between those stations a VC (or PVC if you must) can be created. [so far so good right?]

That case was from A to B, now let's look at the normal case, more stations.

Now A wants to connect to C, A looks in it's routing tables and sees that C can be gotten to from B (I'll spare us all the X.121 addresses) so A does a Call to B with C's address, this sets up a VC from A to B, then B calls the next station (D say) and sets up a VC TOTALLY INDIPENDENT OF WHAT IS GOING ON WITH A.

The limit of 4Ø96 is NOT the number of VC's in the Entire network but the limit of the number of calls BETWEEN two stations (ie A to B)

As far as the routing tables required for X.121 addresses, we would have to have an entry for each area code and also entries for our local nodes The number will be much less than the 4ØØØ entries you talked about.

C2973 CC49 Phil R. Karn (ka9q,2979) 12/1Ø/85 1:Ø8 PM L:13
KEYS:/AMATEUR VC THROUGHPUT/

I have a question for those building X.25-based networks. What will the frame size and packet window limits be? I am concerned about the throughput that will be available through a single virtual circuit.

I am resigned to the development of VC-based networks, so I'm thinking now about how they'll be integrated as subnets of the Internet-based network I'm building. Based on my experience at work with Telenet, considerable gyrations are required (like opening multiple, parallel virtual circuits to the same destination and distributing datagrams among them) because of the tiny packet and window size limits. I hope the amateur VC networks, if built, do not repeat the same mistake.

Phil

C2973 CC5Ø Phil R. Karn (ka9q,2979) 12/1Ø/85 2:41 PM L:52

KEYS:/IS LAPB REALLY NECESSARY?/

It is generally agreed that we need hop-by-hop acknowledgments in amateur packet radio in order to get reasonable performance. It has occurred to me, however, that this does NOT necessarily imply that there needs to be a "connection" between each pair of adjacent nodes.

Current thinking has been to use the full features of AX.25 level 2 to provide a connection-oriented link between each pair of nodes. This path could be used either to support a higher level virtual circuit, or to reliably pass datagrams from one packet switch to another. The amount of complexity in the full AX.25 level 2 has long bothered me. Recently I completed an implementation of the ARPA TCP transport level protocol (the one considered by many to be "too complex" for amateur use) and discovered that it was in fact an EASIER job than my earlier implementation of AX.25 Level 2. I think something's wrong here.

Fortunately, there are "escape clauses" in AX.25 that allow a higher layer to bypass the connection-oriented part (LAPB), namely the UI frame. Ordinarily the UI frame is sent as a pure datagram, i.e., there is no acknowledgment sent by or expected from the receiver. However, the last revision of AX.25 Level 2 specified that should the poll (P) bit be set in a UI frame, a UA response is expected from the other end. It seems to me that this feature provides a way to get link level acks without having to initiate connections and maintain all of the associated state.

Since a packet switch based on AX.25 Level 2 is likely to have many simultaneous connections because of the rich connectivity at the local level, maintaining a large number of protocol control blocks (many of which will be idle most of the time) seems a wasteful use of memory. It seems a better idea when passing IP datagrams from one switch to another to simply send them in UI frames with the P bit set in order to request a link level ack from the other end. The link layer on the sending end must then only hold onto the frame long enough to receive an acknowledgement, or to retransmit the frame until one is received. No memory need be tied up unless there is actual activity on the link, and the LAPB portion of AX.25 need not be implemented.

The only problem with this approach is that there needs to be as many acknowledgment packets as there are data packets. However, recent techniques in congestion control suggest that packet windows be reduced to 1 outstanding packet at a time anyway in order to prevent "congestion collapse" and to avoid clobbering a returning acknowledgement with another data packet. This also avoids the ambiguity caused by the lack of sequence numbers or other indication in the returning UA frame as to which packet is being acknowledged. As long as the packets themselves are allowed to be a reasonable size, good throughput can still be obtained.

Comments or suggestions? I'm going to persue this idea as I continue my TCP/IP-on-AX.25 implementation, and I'd appreciate constructive help.

73, Phil

C2973 CC51 Phil R. Karn (ka9q,2979) 12/11/85 12:19 AM L:186
KEYS:/TCP PROGRAMMERS MANUAL/PLEASE DISTRIBUTE FOR COMMENT/

My work on a TCP implementation for amateur packet radio has reached the point where I can describe the interface provided to the application by TCP. I have also written a UDP (User Datagram Protocol); however, this is not at the same level of maturity as TCP and is therefore more subject to change. This note is meant primarily as "advance information" to any implementers considering writing applications.

To review the purpose of TCP: it supports a reliable, sequenced, byte

[illegible] purpose of TCP. It supports a reliable, sequenced, byte stream "connection" on an end-to-end basis. It fits in roughly at the Transport layer (level 4) of the OSI model. Since a single TCP module supports multiple connections through the use of port numbers, it also provides Session layer (level 5) functionality without the need for a distinct protocol. (Or it makes the session layer unnecessary, depending on your point of view). This package is written as a "module" intended to be compiled and linked with the application(s) so that they can be run as one program on the same machine. This greatly simplifies the user/TCP interface, since it becomes just a set of internal subroutine calls on a single machine. Reliability is much greater, since a hardware failure that kills TCP will likely take any applications with it anyway. Only IP datagrams flow out of the machine across hardware interfaces (such as asynch RS-232 ports or whatever else is available) so hardware flow control or complicated host/front-end protocols are unnecessary.

A TCP connection is uniquely specified by the concatenation of source and destination "sockets". In turn, a socket is the concatenation of a host address (a 32-bit integer) and a TCP port (a 16-bit integer), defined by the C structure

```
struct socket {
long address;/* 32-bit IP address */
short port;/* 16-bit TCP port */
};
```

Therefore it is possible to have several distinct connections established at the same time to a single port on a given machine, as long as the source sockets are distinct. Port numbers are used either through mutual agreement, or more commonly when a "standard" service is involved, a "well known port" number. For example, to obtain standard remote login service (known as "telnet") one initiates a connection to TCP port 23; to send mail using the Simple Mail Transfer Protocol (SMTP) one talks to port 25. ARPA maintains port number lists and periodically publishes them. They will also assign port numbers to a new application on request if it appears to be of general interest.

TCP connections are best modeled as a pair of one-way paths (one in each direction) rather than as a single full-duplex path. Station A may close its path to station B leaving the reverse path from B to A unaffected. B may continue to send data to A indefinitely until it too closes its half of the connection. This is known as "graceful close" and can greatly simplify an application.

My TCP code supports five basic operations on a connection: open, send, receive, close and delete. A sixth, tcp_state(), is provided mainly for debugging. They are summarized in the following section in the form of C declarations and descriptions of each argument.

```
int net_error;
```

This global variable is used to indicate the specific cause of an error in one of the TCP or UDP functions. All functions returning integers (i.e., all except open_tcp) return -1 in the event of an error, and net_error should be examined to determine the cause. The possible errors are defined as constants in a header file.

```
/* Open a TCP connection */
struct tcb *
open_tcp(lsocket,fsocket,active,notify,tos)
struct socket *lsocket,*fsocket;
int active;
void (*notify)();
char tos;
```

"lsocket" and "fsocket" are pointers to the local and foreign sockets, respectively.

respectively.

"active" is 0 for a "passive" open (one in the TCP LISTEN state). A passive open does not cause any packets to be sent, but enables TCP to accept a subsequent active open from another TCP. If a specific foreign socket is passed to a passive open, then connect requests from all other foreign sockets will be rejected. If the foreign socket fields are set to zero, then connect requests from any foreign socket will be accepted. If "active" is 1, TCP will initiate a connection to a remote socket that must previously have been created in the LISTEN state. The foreign socket must be completely specified in an active open.

"notify" is an optional receive "upcall" mechanism, useful when running in a non operating system environment. If "notify" is non-zero, it is taken as the address of a function to be called whenever a "significant" amount of data arrives. This user-provided function may then invoke recv_tcp() to obtain the incoming data.

"tos" is the Internet "type of service" field, consisting of precedence and class of service parameters. There are 8 levels of precedence, with the bottom 6 defined by the military as Routine, Priority, Immediate, Flash, Flash Override and CRITICAL. (Two more are available for internal network functions). For amateur use we can use the lower four as Routine, Welfare, Priority and Emergency. Three more bits specify class of service, indicating that especially high reliability, high throughput or low delay is needed for this connection. The entire TOS field is passed along to IP in each datagram and is interpreted by each IP gateway (packet switch) in the route. The precedence value actually used is the higher of those specified in the two tcp_open() calls.

open_tcp() returns a pointer to an internal Transmission Control Block ("tcb"). This "magic cookie" must be passed back as the first argument to all other TCP calls. In event of error, the NULL pointer (0) is returned.

The only limit on the number of TCBs that may exist at any time (i.e., the number of simultaneous connections) is the amount of free memory on the machine. Each TCB on a 16-bit processor takes up about 129 bytes; additional memory is consumed and freed dynamically as needed to buffer send and receive data. Deleting a TCB (see the delete_tcb() call) reclaims its space.

```
/* Send data on a TCP connection */
int
send_tcp(tcb,data,cnt)
struct tcb *tcb;
char *data;
unsigned cnt;
```

"tcb" is the pointer returned by the open_tcp() call. "data" points to the user's buffer, and "cnt" specifies how long it is. The number of bytes actually queued for transmission is returned; it will equal "cnt" unless some sort of error occurs, such as lack of memory for buffering. The data is copied into an internal queue, a transmission is attempted, and the function returns so that the user may immediately reuse his buffer. TCP uses positive acknowledgments and retransmission to ensure in-order delivery, but this is largely invisible to the user. TCP enforces no limit on how much data can be queued for transmission, so the user should be careful not to run the system out of free memory. (This is something else that requires a multitasking kernel to do right).

```
/* Receive data on a TCP connection */
int
recv_tcp(tcb,data,cnt)
struct tcb *tcb;
char *data;
unsigned cnt;
```

The arguments to recv_tcp() are identical to those of send_tcp(), except that

The arguments to recv_tcp() are identical to those of send_tcp(), except that any data on the connection's receive queue is placed in the user's buffer, up to a maximum of "cnt" bytes. The actual number of bytes received (the lesser of "cnt" and the number pending on the receive queue) is returned. Since this TCP module cannot assume the presence of sleep/wakeup primitives provided by an underlying operating system, recv_tcp() is currently designed to return -1 with net_error set to EWOULDBLK if no incoming data is pending. The "notify" feature on open_tcp() is provided to eliminate the need for constant polling of the recv_tcp() function; whenever TCP calls the notify function, it guarantees that recv_tcp() will not return -1. (Technical note: "notify" is called whenever a PUSH or FIN bit is seen in an incoming segment, or if the receive window fills. It is also called before an ACK is sent back to the remote TCP, in order to give the user an opportunity to piggyback any data in response.)

When the remote TCP closes its half of the connection and all prior incoming data has been read by the local user, subsequent calls to recv_tcp() return 0 rather than -1 as an "end of transmission" indicator.

```
/* Close a TCP connection */
close_tcp(tcb)
struct tcb *tcb;
```

This tells TCP that the local user has no more data to send. However, the remote TCP may continue to send data indefinitely to the local user, until the remote user also does a close_tcp(). An attempt to send data after a close_tcp() is an error.

```
/* Delete a TCP connection */
delete_tcp(tcb)
struct tcb *tcb;
```

When the connection has been closed in both connections and all incoming data has been read, this call is made to cause TCP to reclaim the space taken up by the TCP control block. Any unread incoming data is lost.

```
/* Dump a TCP connection state */
tcp_state(tcb)
struct tcb *tcb;
```

This debugging call prints an ASCII-formatted dump of the TCP connection state on the terminal. You need a copy of the TCP specification (ARPA RFC 793 or MIL-STD-1778) to interpret most of the numbers.

Well, that's it. Constructive comments on this interface are welcomed.

73, Phil Karn, KA9Q

C2973 CC52 Phil R. Karn (ka9q,2979) 12/11/85 10:32 PM L:60
KEYS:/TCP/IP FOR MACINTOSH INFO/

Jack (and others), I've obtained the Internet package for the MacIntosh that I mentioned earlier. It runs a little over half a meg with sources and binaries (in encoded ascii form). I know nothing about this package other than what you read here, which is the original announcement.
 Phil

I have posted the sources and binaries for the Macintosh Internet Protocols to net.sources.mac . The sources run about 12,000 lines of code, and another few thousand lines of binhex'd binaries. The source and binaries are stored in twelve shell archive files; note that "sh" and not "csh" should be used to unpack the archives, and that you should do the unpacking in a new, clean directory. The first archive file absolutely must be unpacked first; after that, the order doesn't matter.

The Internet Protocols are network communication protocols defined by the

Department of Defense, in wide use around the world. The Macintosh versions run on Appletalk, the low-cost local area network for the Macintosh computer (and, using a Centram board, for the IBM PC). They are meant not so much for Macintosh to Macintosh communication as for high-speed, error-checked communication with minicomputers and mainframes, such as VAXen and TOPS-20 systems. Some form of Seagate-compatible router will be needed in order to use these programs as they are intended to be used. (Seagate, the Stanford Ethernet Appletalk Gateway, is a family of routers defined by compatible software; the Seagate source code may be retrieved by FTP from SUMEX, in the INFO-MAC archives. Those of you who are not able to FTP should send mail to Bill Croft, su-safe!croft, for information on the tape distribution.)

The programs included are TFTP (simple file transfer), TELNET (remote terminal emulation), CUST (to customize your Macintosh's IP address and so on), and SFMTEST, a test program provided as an illustration of the multiple file "Get File" dialog. Both sources and binaries are provided for all these programs.

The sources may be of interest even to people who have no use for Appletalk, because they contain software for emulating DEC vt100 terminal graphics, error reporting to the user, running multiple tasks simultaneously (actually synchronously) within an application, setting timers at a higher level than direct manipulation of the vertical retrace queue, and an extended "SFGetFile" (standard file package "Get File" dialog) that can be interfaced to from any language without translating the source code, and which allows multiple files to be selected. The sources are in Lisa Pascal, but Lisa Pascal can readily be transliterated into any of the Mac C compilers.

The code originated with the MIT PCIP network package for the IBM PC, written in C. Initial translation was done by Mark Sherman of Dartmouth; I continued this work. The final product is far more than a translation, being thoroughly integrated into the weird and wonderful Macintosh environment, and incorporating a set of, oh, roughly aleph-null bug fixes and major extensions.

Second draft user-level documentation may be found in the file "user" in the first file of the twelve files in the posting. Better documentation is currently being written; by someone else, since I have moved on to other projects, such as LaserWriter print spooling on a UNIX machine. There is currently no external documentation of the source files, though they are reasonably well commented.

-=-

Tim Maroney, Professional Heretic, CMU Center for Art and Technology
tim@k.cs.cmu.edu | uucp: {seismo,decwrl,ucbvax,etc.}!k.cs.cmu.edu!tim
CompuServe: 74176,1360 | God is not dead; he just smells funny.

C2974 CC109 Phil R. Karn (ka9q,2979) 12/ 9/85 2:45 PM L:216
KEYS:/IF YOUVE GOT THE NICKEL/

Howie, I think we have some misunderstandings about what we mean by certain terms when discussing network architectures.

"Command processor" was probably not the correct word to use. What I was referring to is any software module whose function it is to act apon messages from a "client" and possibly respond to them in some way to provide "service". If the "client" is a human sitting at a keyboard typing ASCII strings and parameters, this is what you usually think of as a "command processor". However, the "commands" might be coming from another computer, in which case you might want to design the "command set" (both syntax and semantics) in an entirely different fashion. Of course, a "command processor" that talks to a machine instead of a human might better be called a "protocol module", but then again a module that understands ascii strings typed by humans really IS implementing a "protocol" as well.

What I'm trying to say here is that "command interpreters" and "protocol

What I'm trying to say here is that "command interpreters" and "protocol implementations" are conceptually the same thing: two processing entities, with or without state, that exchange messages according to some rules (i.e., protocols). In this sense, a TNC command set is a full blown protocol. It is a well established fact that protocol designs that involve a large amount of state (memory) tend to be much more complex, difficult to implement and to debug than those that maintain little or no memory between messages. The problem often becomes overwhelmingly complex when more than two entities are involved and they must all maintain consistent views of each other's state across lossy communication paths. Minimizing the number of places where state must be kept is a practical necessity. Managing virtual circuits across a series of switches, passing call setup and teardown messages back and forth, and handling exception conditions is clearly much more difficult than simply routing packets in stateless switches and confining the notion of a "connection" (if needed) to the endpoints. Just yesterday at the Digital Committee meeting I watched Eric Scace go through a single call setup and teardown operation in Q.931 (the network protocol for ISDN). His flowcharts spanned 16 pages, with 54 (!) messages flying back and forth. TCP's state diagram fits on a single slide, and it only runs at the endpoints of a network, not in the middle. Three end-to-end messages get you connected, and three end-to-end messages get you disconnected.

I'm sorry you feel that TCP/IP is "high flake intellectual crap" from the ivory towers, because there exists strong evidence to the contrary. An operational Internetwork consisting of dozens of different network types (including Packet Radio) and literally tens of thousands of hosts of all flavors that all manage to interoperate with each other has existed for almost five years. That section of the Internet within my own company consists of several dozen heterogeneous networks at several locations, supporting several hundred hosts used on a day-to-day basis to do real work that we get paid real money for. If that isn't "real world" enough for you, there are implementations of the Internet protocols available for just about every major operating system, and commercial vendors sell these for real bucks.

People buy them because they WORK. Not overhyped, years-away-from-here "vapornets" like ISO and MAP and ISDN, but stuff you can buy off the shelf NOW, plug in, turn on and use. Much of this is the result of TCP/IP being a DoD spec, but many organizations (such as my own) who have nothing to do with the government use it anyway because it works just as well for our needs.

I agree, I don't care much about memory space because memory is cheap (unless you run into a hard limit like 64K address spaces). But program size also tends to be program complexity, and that's certainly something I'd like to minimize. I asked for the TNC-2 programming model because my next step is to port my TCP to the TNC-2. Others are already busy porting it to a variety of machines, and applications people are looking at putting "BBS-like" functions on top of it. Next task is to port some full-blown IP packet switch code I have to the 82Ø and TAPR NNC.

I fail to understand your contempt for the UI frame. It makes perfect sense when you view AX.25 for what it really is - a two-layer protocol with a VC-oriented pseudo-transport protocol (LAPB) on top of a source-routed pure datagram protocol (the address fields). The UI frame is simply a way to bypass LAPB and get directly at the datagram sub-layer. Nobody forces you to use it, but it's extremely handy when you need a broadcast mechanism -- something that is difficult or impossible to implement efficiently on top of a VC interface.

Regarding transport protocols - your statement is a non-sequitor. Transport and link level protocols do separate jobs, and both are needed even if one of them turns out to be trivial. The real issue is how the various FUNCTIONs need to be allocated among the layers. My view (which is widely shared -- see Salzer's paper "End-to-End Arguments in System Design) is that the end-to-end level (i.e., the transport level) is the

design) is that the end-to-end level (i.e., the transport level) is the ONLY place where you can begin to guarantee "correctness" (i.e., reliability); anything you place at the lower levels (e.g., link) can be justified only on a performance basis. In many situations, attempts to make things more "reliable" at the link layer are redundant and just waste of bandwidth. Packet radio is about the only exception I can think of where link level reliability measures are justified on a performance basis. In virtually every other situation, links are either "up" (with a 10^-12 BER) or "down" (with .5 BER), so it isn't worth the effort.

If you really understood the Internet approach, you would realize that you can put almost ANYTHING (including VC networks) down below the Internet protocols. My horizons extend further than just AX.25 Level 2, and there are lots of other networks out there that I want to interoperate with transparently: local datagram networks (Ethernet, Pronet, packet radio) or long haul virtual circuit networks (ARPANET or X.25). It makes no difference; a waxed string and a pair of tin cans are suitable if they can relay a datagram with some reasonable degree of reliability from point A to point B. These networks can and do coexist quite nicely in the Internet, because the Internet Protocol provides a uniform, standard service that hides the "seams" between these dissimilar networks for the higher level protocols while at the same time demanding as little as possible from the underlying network protocols.

So Howie, if you want to build a VC based network, fine! I certainly won't stop you, because IP can still use it. I'll nail up some virtual circuits through your network between my IP gateways and run IP datagrams over them. We do exactly the same thing with a Telenet X.25 connection right now, and that's how we access the rest of the ARPA Internet. Plus I can provide connectivity between your network and any other type of network people might want to experiment with. So it's really silly to argue; with IP we can all coexist and intercommunicate at the same time.

End-to-end acknowledgments are not provided at the Internet level; they are strictly the province of the transport protocol, which is a bilateral agreement involving only the endpoints of the "connection". Nobody requires you to use TCP if your application doesn't want or need end-to-end acks. You are free to develop your own transport protocol if you wish; the Internet couldn't care less. Link level acks aren't specified either; if the lower level protocol wants to provide them, that's transparent to IP also. Since the Internet layer is the only layer that everybody IS required to implement (in order to be interoperable), it was deliberately made as simple as absolutely possible -- hence the datagram-only service, since it's the least common denominator.

I don't understand your question about flow control. Conceptually TCP does per-VC flow control just like the packet layer of X.25 (which isn't a transport protocol, by the way, because it isn't truly end-to-end). Both use end-to-end (well, ALMOST end-to-end for X.25L3) windows and acks to control how much data can be in transit at any time. Other than where the protocols sit and how robust they are against network faults, they behave similarly.

From my own comparative experience, TCP, easily the most complicated single member of the Internet "suite", was considerably easier to implement than AX.25 Level 2. The IP layer/packet switch code is much simpler than X.75 (since there are no connections) and the only real complexity (needed in both our approaches) is in the automatic routing algorithms.

Regarding your question on congestion control: there are several aspects to the problem.

1. First, it is the end-to-end transport protocol that decides when to send a datagram, so it needs to be reasonably careful about doing this efficiently. For example, it is technically legal by the TCP spec to fill

up a 2,048 byte receiver window with 2,048 1-byte TCP packets although it is clearly stupid to do so. A convention has been established in TCP implementations where only 1 packet will be allowed "in the pipe" at any one time, unless enough additional data can be sent to fill out a full-sized packet (in which case you don't save anything by waiting for more from the user). This technique works extremely well because it allows the transport protocol to put more user data in each packet without also slowing things down when the user needs lots of bandwidth.

2. Contrary to popular impression, the Internet Protocol is NOT a "pure datagram" protocol in the sense that a digipeater network is (i.e., send a datagram and pray, never getting any network-level indication when something is wrong). There is a mandatory adjunct to IP called ICMP (Internet Control Message Protocol) which reports errors to the sender of a datagram. If congestion occurs (or is about to occur) in an IP switch, it is required to send back one particular ICMP message called a Source Quench. This tells the transport protocol (e.g., TCP, but there could be others) to back off; the network is getting congested. Only if the user ignores Source Quench will the switches finally be forced to toss stuff on the floor. No ICMP message is sent unless there is a problem. Since most datagrams make it without a hitch this avoids a LOT of redundant overhead packets that just keep reassuring you that all is well with the network.

I think you'll find that virtual circuit network protocols are no panacea for congestion control, because congestion is always possible wherever you have shared resources. The only way to guarantee bandwidth during a connection is to abandon packets and use circuit switching; the only way to absolutely guarantee that you'll get a circuit when you need one is to put in a leased line. Whenever you share something, you can get congestion.

Yes, there are two fundamentally different approaches to building networks. But one approach (that of the Internet) is fundamentally superior if you're trying to support a loosely- (or un-)coordinated set of experimenters with a wide variety of top-level applications and lower level protocols but still wishing to intercommunicate.

You can't prove anything with metaphors. I admit I like them too, so I guess I'll add mine.

Unlayered virtual circuit nets are like railroads - track is expensive, but it provides a standard, high quality path. Virtual circuit transport protocols are therefore like railroad locomotives - good on fuel economy/ton but very poor on flexibility. They are highly specialized devices that only run on steel rails, and there are lots of places that steel rails don't go. VC nets and railroads work best when doing one very specific, customized application on a large scale where the lack of flexibility isn't a problem, like hauling coal from mines to powerplants or connecting remote dumb terminals to big IBM machines.

A datagram net is more like the road system. Some are Interstate highways, some are dirt roads, some are in between. They're owned and maintained by thousands of different organizations and private individuals. All are interconnected so I can almost always get from here to there. A transport protocol designed to operate on a datagram net is like an all-terrain vehicle - a general purpose device that can drive over almost anything. And you can still put it on a flat car and ship it by rail. (At the risk of really stretching the metaphor, my vehicle also has wheels so it can drive on rails, but the railroad forces me to rent a flatcar. Just like the X.25 carriers force me to buy extra from them, i.e., a virtual circuit, when all I want is a raw datagram service.)

Granted, the fuel economy of an all-terrain vehicle might not be as good as a locomotive per ton hauled. But let me ask you this: which would you park in your driveway as your SOLE means of transport?

Phil

C2974 CC11Ø Howard Goldstein (N2WX,2987) 12/11/85 9:49 PM L:39
KEYS:/QUENCH/SOUNDS LIKE A NEW SOFT DRINK/
A: 1Ø9

Phil,

I'm more concerned about the people reading this conference who didn't expect to see a 216 line reply! In the interest of brevity - seemingly an impossible goal in these discussions - can we confine this discussion to NETWORK and related issues? Don't care if it's interface or internet. Moreover, the sooner matters of semantics are dispensed with, the more meaningful dialogue can result.

With this in mind...

No one argues the utility of a broacast UI, but (you must know this!) the new UI in ax2512v2 does that and SO MUCH MORE. Alas, what "more" is supposed to be isn't clearly defined. That lone paragraph taken in the context of the rest of the document is vague.

Run anything on top of a vc net? Sure! Gateways? Excellent! And thank you for yielding on the requirement that EVERYONE run TCP.

This ax2513 bashing continues. Flow control in 13 is not almost end to end, it is ALWAYS hop by hop. The window is NOT preserved end unlike TCP WHICH IS TOTALLY END TO END.

Too, messages may be nested within themselves, for example an 13 ack can also ack at 12 in the same instant. Wish Eric were here to respond to your comment. Were 54 individual - in time and space - messages exchanged end to end?

I promised not to discuss transport issues but I thought you had proposed to make it's embellishments -- like ICMP -- mandatory throughout the network; not only at the endpoints. Anyway it seems now inextricably (sp) tied up with functions which are really the providence of lower levels, as with the matter of congestion control 'cause what happens when the quench - or other 'transport' units get lost?

Once again friend, it's End to End.

Non-sequitor? Hardly. Doesn't it depend on what "flavor" TCP you're running?

Did I hear somewhere that someone was running TCP over the phone lines? Guess a modem could be considered part of the link-layer. Possibly some of that group belongs to the other one that - conceptually, now - considers digipeaters the domain of the link layer. A NULL link layer is the same as one that isn't there - this has the same affect as doing away with same, and that's what was inferred.

Gosh this is long! Will pass on the mandatory methaphor despite the enjoyment everyone gets out of them.

Howie

C2974 CC111 Phil R. Karn (ka9q,2979) 12/12/85 12:19 AM L:93
KEYS:/AX25/SOUNDS LIKE A NUMBERING SCHEME FOR SLASHER MOVIES/

When people tell me that my comments are too long, I'm reminded of two famous quotes. The first is by Abraham Lincoln, who when asked how long someone's legs should be, replied "just long enough to reach the floor." The second is Albert Einstein's famous saying that "things should

floor. The second is Albert Einstein's famous saying that "things should be simplified as much as possible -- but no simpler."

The issues of Network (3) and Transport (4) are so closely intertwined that it's impossible to discuss them separately. Indeed, it is the interface between these two layers (and which functions ought to go in each) that is so controversial. The basic position to which I subscribe is that certain functions (such as reliable end-to-end delivery and flow control) can ONLY be done properly at the transport layer. Therefore, attempts to do them at lower layers are at best duplicated efforts and at worst add enormous unnecessary complexity and performance bottlenecks.

I feel that we have made a major mistake in amateur packet radio by working upwards from the link layer; if we had defined a transport protocol FIRST and worked down, we'd have encountered this fact of life a lot sooner. Instead we're using AX.25 Layer "2" as though it were a transport protocol, and when we do finally come out with a "real" transport protocol all the users will have to change their habits (or go through Magic Protocol Conversion Gateways). KA6M first proposed this alternate strategy over FOUR years ago, and I really regret that Hank doesn't have a louder voice. I quote from his letter of June 1Ø, 1981 to Paul Rinaldo, W4RI:

"...it [the Internet protocols] would save us an enormous amount of development time in bringing up our network. I'm really afraid that if we started with something much simpler, as some hams will encourage us to do, we would find in a year or two that features are lacking and extensions are necessary. Much discussion and haggling would produce a new and revised spec, and piece by piece we would eventually reinvent most of the features that are currently in the Internet document..."

As I said, the use of TCP as a transport protocol is a convention observed by the end hosts; the network doesn't care. However, it is THE protocol used when an end-to-end virtual circuit service is needed, so you are not likely to be able to talk to many services unless you have it. That's why I'm concentrating my time on an implementation which I am giving away to the amateur community.

I probably don't understand your implementation. I was assuming earlier that you implemented your level 3 in the manner described to us a year ago by Eric Scace, in which layer 3 flow control packets are passed unchanged (except for the translation of VC numbers) across X.75 gateways from end DTE to end DTE. He explained that this is how Telenet operates internally. Layer 3 flow control is therefore "almost" end-to-end, or, in other words, the network behaves as though the "D" bit is always on. While I've cursed at them for this, it is only because of the tiny window values they chose. If you are instead relying solely on link-level "backpressure" then I think you're going to be in serious trouble. As soon as people start shoving big files into the network without a transport protocol, they'll likely "gridlock" every TNC in the path. So they'll still need some sort of end-to-end transport protocol that limits the number of outstanding packets in the network, and you may also require a "source-quench" mechanism in case too much traffic from different sources happens to pass through a bottleneck.

I was just counting the sendmsg/recvmsg pairs in the Q.931 SDL diagrams. If you are on the Digital Committee mailing list, you will most likely receive a copy. These represented the transactions between originator and transit switch, and transit switch and destination. They were not all end-to-end.

ICMP is implemented as a "pseudo-transport" protocol in that it has an IP protocol type assignment as does TCP, but it is really better thought of as a part of IP. ICMP messages are generated by gateways (IP packet switches), which do not have to implement any other protocols like TCP (unless they are also hosts, of course). ICMP messages are purely ADVISORY. A transport protocol running atop IP has no guarantees that all its datagrams will be delivered, nor is it guaranteed an ICMP message should a datagram be dropped (ICMP messages are themselves datagrams, and datagram delivery is never guaranteed).

are themselves datagrams, and datagram delivery is never guaranteed).
Therefore, a transport protocol such as TCP must still use positive acknowledgments with retransmission to assure reliability. ICMP messages only help to speed things up, or to help diagnose a network problem. For example, if TCP receives an ICMP "destination unreachable" when trying to establish a connection, it doesn't have to retry N times before giving up; it can notify the user right away. If the network becomes congested, TCP will likely encounter higher packet loss rates and begin backing off its retransmissions anyway; an ICMP source quench just starts the process early, before the network is forced to resort to dropping packets.

I don't understand your last paragraph, the one about modems being part of the link layer. Actually, I think you'll find that although real networks are layered, the pieces seldom conform in a nice one-to-one fashion to each of the 7 ISO layers. The OSI model is supposed to be a rough, descriptive GUIDE for understanding existing networks, not a mandatory straightjacket for implementers. There is a cute "proof" in one of the USC-ISI papers that, according to the ISO model, N = N + 1 for any value of N. The point they were making is that layers are relative. Even the Internet protocols don't go far enough, in that I believe real networks are becoming multi-level hierarchies of sub- and super-networks, and the original Internet model only allows two (subnet and internet).

Phil

C2974 CC112 Howard Goldstein (N2WX,2987) 12/12/85 10:39 PM L:19
KEYS:/AND VC'S ARE A LOT OF FUN TOO/
A: 111

Phil,

My implementation supports the "D" bit OFF only - not a major decision really because `almost end-end' acknowledgements, D=true, is specified as an OPTION to be negotiated at call setup time (in X.213 network service specs) and define hop-hop as default.

Similarly, "backpressure" is applied on a HOP BY HOP basis, at or before the time translation takes place to the other LCN. Since the links themselves never get "stopped up", they don't gridlock.

TELENET's decisions don't have to make technical sense. How they implement has as much to do with ham packet as it does with concern for elegance and flexibility. On the other hand, we're Amateurs having one hell of a great time and are unconcerned with overhead and capital cost.

I'm not sure how my example of a modem plus "TCP and the sound of electrons" as kind of a null level 2 was understood as a mandate to try to implement the model in contradiction to its own directive NOT to understand it as a mandate....

C2974 CC113 Phil R. Karn (ka9q,2979) 12/12/85 11:27 PM L:33
KEYS:/BUFFERING AND GRIDLOCK/

Howie,
I believe the reason Telenet did it the way they did is to guard themselves against congestion by limiting the number of outstanding packets on any single virtual circuit. Since the predominant use of their network is for slow speed remote terminal access, the throughput limit this creates is not a real problem. It *IS* a major hassle when doing host-to-host file transfers, though, which is why we go to such lengths to circumvent it. Of course, our technique for circumventing it runs directly counter to their reason for doing it in the first place, and since they don't have a "source quench" mechanism to tell when their network REALLY gets into trouble, there's nothing we can do to help. I wouldn't mind helping out if they didn't ask us to cut back unless it was really necessary, but I do object to having lousy throughput all the time.

necessary, but I do object to having lousy throughput all the time. It's like not being allowed to carry on more than two pieces of luggage on board a plane, even when it's half empty and there's plenty of space.

How do you do link level flow control when handling multiple simultaneous links? Unless you do it by static allocation of X amount of buffer space to each link connection (whether or not it is used) you might encounter some rather nasty gridlocks, e.g., where A can't send to B because B's buffers are filled with traffic to C, who can't accept them because its buffer's are filled with traffic to A, and so on. If you do dynamic buffer allocation (necessary to handle a large number of links on a small machine) and don't take extra precautions, especially when traffic might sit in a given machine indefinitely, then I think you're leaving yourself open to deadlock.

I see that you have two choices: either use dedicated buffers (and deny additional links, even if the existing ones are idle) or allow many more links with dynamic buffer sharing, and take the risk of having to drop a packet occasionally when things get busy.

Phil

C2974 CC114 Phil R. Karn (ka9q,2979) 12/13/85 1:38 AM L:7
KEYS:/CONGESTION CONTROL/

Howie, if you have a copy of Tanenbaum ("Computer Networks") take a look at the section on Congestion beginning at page 215. He provides a pretty complete description of the buffer allocation and deadlock problems, along with their most common solutions. Using his terminology, the ICMP Source Quench is a "choke packet".

Phil

C2974 CC115 Howard Goldstein (N2WX,2987) 12/13/85 8:05 AM L:23
KEYS:/BUFFERING AND GRIDLOCK/HOW MUCH OF AN IMPROVEMENT IS IMPLEMENTATION DEPENDANT/
A: 113

Phil,

Granted that if it was done, flow control solely through the link level would cause the gridlock problem you describe.

But flow controlling at that level is NOT the desired end, I don't think. At least that's not what I'm shooting for.

RNRing the link is prevented directly at the link level by insuring that enough buffer space exists to process a level 3 packet. If packets are processed individually, which they are, this buffer need only be the size of the largest packet one will ever expect. Link flow is no longer an issue.

Now let's get down to some (implementation specific) nitty gritty. Static allocations of X amount of buffer MUST take place at call setup time and for each hop in the network - but X can be a very small number, maybe 20 or 30 bytes, since we're only concerned with preserving the state of the level 3 interface, and are never forced to accept data units.

How a participant in the network arranges to allocate for network data units is IMPLEMENTATION DEPENDANT and subject to local change through improved hardware or whatever.

I'm not claiming VCs are any panacea, only that through their use we can build a highly usable network with the restrictive development and destination tools available today (eg. 820) yet still be prepared to offer even more in the way of throughput and services by way of the better processors and/or greater memory space we'll have tomorrow.

C2974 CC116 Phil R. Karn (ka9q,2979) 12/13/85 8:42 PM L:29
KEYS:/BUFFER MANAGEMENT/

Howie, maybe it would help if you described your implementation in more detail, so I can see how you solve the problem. It's NOT just an "Implementation Dependent Detail", because certain approaches require features to be in the protocol that might not be there.

You're right, static allocations can only be done with virtual circuit protocols, but that's not the only way to do it. Static allocation has the advantage (if done right) of completely eliminating all possibility of deadlock, but it does so by turning a "packet" network (where things are supposed to be allocated on demand) onto a "pseudo-circuit switched" network, where resources are busy whether or not real traffic is flowing. To me, this isn't true "packet switching". If you want to build a circuit-switched network, fine; they have plenty of appropriate applications, but bursty, interactive data networking isn't one of them.

The only way you can guarantee your static allocations is by refusing additional connections when you run out of memory, even if your switch hasn't seen a packet for three days. You may want to consider this point, for no other reason than that I will be coming along and nailing up virtual circuits over which to pass datagrams, and I don't want to have to bother with timing out unused circuits and re-establishing them on demand.

One of the things I like being able to do is to stay logged into a remote machine all weekend without having to log out, because I consume absolutely no network resources by doing so. This is not the case with VC subnets, at least not without considerable complications with session-layer "reconnects" and other similiar stuff.

C2974 CC117 Howard Goldstein (N2WX,2987) 12/14/85 9:33 AM L:2Ø
KEYS:/AND NOW FOR SOMETHING COMPLETELY DIFFERENT/REAL SYSTEMS/
A: 116

Phil, your statement in re: implementation dependent detail is non-sequiter. Recall there is a minimum, mandated functionality which is set forth in the osi network service definition.

Are you not stretching it a little bit on allocations? When an IP net drops packets, it might or might not get a message back to the endpoint through an indeterminate # of nodes saying "shut up." And then more and more resending until the throughput approaches the zero point.

It makes a lot more sense to shut up a neighbor than to shut up an endpoint thousands of miles away.

I feel its worth 2Ø-3Ø bytes per call to do it right (i.e., locally). Besides, memory is cheap (right?), plus you run into throughput constraints way before you'll run out of 2Ø to 3Ø byte blocks. I figure at that rate you run out of static allocations after something like 3ØØ calls/interface.

If we've come to arguing over minor turds like this, then the dialogue has surely been worthwhile.

Oh, yes, if you are concerned about timouts then why don't you just treat the network as a big datagramme system and pump restricted response call requests through the damn thing? I see we can argue like this for many more months - perhaps years; But I want to get on with implementing real things.

C2974 CC118 Phil R. Karn (ka9q,2979) 12/14/85 7:39 PM L:52
KEYS:/CONGESTION CONTROL/

>Recall there is a minimum, mandated functionality which is set forth in the
>osi network service definition.

The ISO RM is supposed to be a GUIDE for describing existing networks, not a mandatory set of requirements for implementing them. Attempts to do otherwise belong in the same category as attempts to legislate the value of pi.

An IP gateway may choose to return an ICMP Source Quench when it sees trouble brewing, in advance of actually having to drop packets. If the host properly obeys the protocol, then there won't be "more and more resending until the througput approaches the zero point". It is true that in a datagram network an improperly functioning transport protocol module (one that ignores source quenches and/or the far end window) and floods the network can cause problems, unless a nearby gateway notices the problem and takes appropriate action (like dropping all its packets). However, similar things can happen in a VC network. What do you do with a station that decides to flood the network with call setup requests? Congestion control is a much more fundamental problem than the choice of network protocol.

>It makes a lot more sense to shut up a neighbor than to shut up an endpoint
>thousands of miles away.

It makes more sense to me to have an end-to-end protocol that keeps packets out of the network unless it seems likely that they'll be delivered quickly to (AND accepted by) their destination. Just like a highway, a packet network's primary job is transportation, not storage. You're perfectly free to use link-level flow control on a datagram network. We do that but have found that it doesn't buy much. It usually just gets in the way by generating twice as many interrupts per second that the CPU has to process.

I'm confused by just how much you allocate per VC. Obviously you need a control block (20-30 bytes) but aren't you also pre-allocating your DATA buffers? If instead you work out of a shared buffer pool, you MUST have rules that intentionally drop packets whenever necessary to ensure having buffer space for incoming link level acks that would free up other packet buffers. Otherwise you leave yourself open to deadlock. Consider the following scenario, known as "store-and-forward deadlock":

Station A has N packets for station B, but has no buffers left to accept an acknowledgment from station B. Station B has N packets for station A, but has no buffers left to accept an ack from station A.

Plenty of variations exist; you could have a ring of stations instead of just two all waiting for events that never happen.

I *could* use your network in "fast select with restricted response" mode, but I suspect it'll perform very poorly. I'd find it easier just to nail up a circuit so I don't have go invoke all your setup overhead on each and every datagram packet.

Phil

M 23591 Lyle Johnson (WA7GXD,2973) 12/16/85 11:16 PM L:19
KEYS:/TNC2 REV2 NOTICE/PLEASE POST ON PBBS'S/
TO: WA7GXD, (Group 95)

To: All TNC 2 Rev 2 Owners
From: TAPR
Re: Minor Hardware Bug

The Rev 2 PC board contains a new resistor, R98. This resistor provides a software-readbale input directly from the modem for future use with AMTOR, RTTY and the like. It is supplied as 4.7k ohms.

Unfortunately, the SIO chip used at U22 provides a pulse output on the same pin that R98 is attached to when the SIO is running in HDLC mode (as it does on packet).

The result is possible interference with State Machine operation.

Please change the value of R98 to 100k ohms for proper packet operation.

Thank you.

C2974 CC119 Howard Goldstein (N2WX,2987) 12/16/85 12:18 PM L:23
KEYS:/CONGESTION/SEMANTICS/X.213/
A: 118

A TCP/IP free for all network flooded the way you describe it would approach zero throughput a lot faster than a VC type. As congestion builds, or with low reliability links slowing everything up, end-end timers of TCP are invoked and more flooding ensues. Chaos prevails until connections are forced down.

A VC switch can simply reject new calls it can't handle if it's being flooded, passing the onus to the preceeding switch or the calling transport layer for redirection.

Won't an approach where the responsibility for congestion control at an interface is done locally, by the interfaces in question, work best simply because the end users of the network don't have to get involved in remote clogging issues?

I don't understand how the truism that ISO RM is a model has anything to do with the (out of context!) quote taken:

>"Recall there is a minimum, mandated functionality which is set forth in the osi network service definition."

This most clearly refers to X.213, a "NETWORK SERVICE DEFINITION FOR OPEN OPEN SYSTEMS INTERCONNECTION FOR CCITT APPLICATIONS" - a SPECIFICATION, and not a ```guide for describing existing networks''' (although I wouldn't be surpirsed if it had a part in aiding some design open systems).

Whether it falls into the category of ```attempts to legislate the value of pi''' is as much an argument against a TCP/IP/ICMP spec as it is AX.25 or any other...

C2974 CC120 Phil R. Karn (ka9q,2979) 12/16/85 1:14 PM L:50
KEYS:/CONGESTION/ISORM/

I'd like to send you a copy of ARPA RFC-896, "Congestion Control in IP/TCP Internetworks," by John Nagle of Ford Aerospace. He has come up with some ideas that are very good, and have virtually eliminated the phenomenon you describe. My TCP uses his algorithm (which limits the number of outstanding packets on a connection to 1 unless there is enough data to generate more

maximum-sized packets), plus it has rround-trip timing with an exponential backoff strategy so that it 1) adapts its retransmission timer to the actual time needed to acknowledge a segment and 2) backs off very rapidly on successive retransmissions when things don't seem to be getting through. These algorithms combined do very nicely when working between two Ethernets connected by a slow (1200-9600 baud) link.

The problem with your technique for congestion control is that it rejects calls even when the packet switch involved hasn't seen a data packet in three days because it is filled with idle virtual circuits. Besides, even limiting the number of virtual circuits won't prevent deadlock unless you've made static buffer allocations to each VC, and at the rate of a few K per VC you run out pretty quickly.

I don't care what illusions of grandeur the ISORM-ites (term attributable to M. A. Padlipsky) have, their model is only that -- a model. Consider the following quote from "The ISO Reference Model and Other Protocol Architectures" by Postel and Cohen:

"We accept the ISORM as a great contribution for teaching about the underlying concepts of computer communication. On the other hand, we cannot accept it as a model for all seasons, as an implementation specification to be followed to the letter in all situations.

"Note the striking similarity between these statements:

* In 1978, ISO invented the ISORM; this made computer communication much easier.

* In 1777, Lavoisier invented oxygen; this made breathing much easier.

"We do not attribute such statements to the OSI committee, but to camp followers who have misunderstood the ISORM and its significance. Just as Lavoisier's invention of oxygen did not change breathing, the ISORM did not change the approach to communication. At best, it helped some people understand it better.

"According to Babor and Lehrman, oxygen was first discovered by Joseph Priestly, who isolated it from mercuric oxide and reported it on August 1, 1774. In October 1774, Antoine Laurent Lavoisier started repeating Priestly's experiments. After three years of work, Lavoisier published his report on oxygen, for which he claimed eternal credit. As noted above, the similarity to the ISORM situation is striking."

Phil

C2974 CC121 Howard Goldstein (N2WX,2987) 12/17/85   7:39 AM L:6
KEYS:/PADLIPSKY POSTEL AND COHEN/ABRAHAM MARTIN AND JOHN NEXT?/
A: 120

Padlipsky, Postel, Cohen...

Have these guys done anything for amateur networking?

The analogy was enjoyable, but - as the Georgia SOUTHNET folks may be fond of saying - you can't put all those papers, even combined, on a mountaintop and make them work.

C2974 CC122 Phil R. Karn (ka9q,2979) 12/17/85   1:49 PM L:28
KEYS:/WHAT HAVE THEY DONE FOR US?/

Well, let's see...
From Gateway, Vol. 1, No.4, Sept 25, 1984 (that's over a year ago):

"We received the following letter from Richard Bisbey II, NG6Q:

"'The Information Sciences Institute Amateur Radio Group (WB6MXZ) has been on the air with a 56 Kb packet radio system since December, 1983. The equipment consists of 8Ø88-based controllers using the Zilog 853Ø SCC and frequency agile 1Ø-watt FSK transmitters and receivers. The controller supports serial, parallel, and Ethernet connected devices. Assembled controllers cost approximately $65Ø. The system uses standard INTERNET (TCP/IP) protocol, and in addition to supporting simple link level connections, supports internetworking as well as numerous high-level protocols such as TELNET, File Transfer (FTP), Mail (SMTP), Multi-Media Mail (MMM,MPM), Graphics (GP), Packet Voice (NVP) and Remote Virtual Disk (RVD). The hardware and software can support data rates as high as 1 Mb well in excess of the current amateur limit of 56 Kb..."

They go on to talk about a transcontinental internet test from an airplane, and a demo of intercontinental "internetting" from a packet radio in California to Europe via satellite. They're also working on a low-cost spread spectrum version of the system.

We don't hear much from these guys mainly because they see the rest of us as hopelessly stuck back in the Stone (well, maybe Bronze) age, and they're absolutely right.

Phil

C2974 CC123 Lyle Johnson (WA7GXD,2973) 12/17/85 9:19 PM L:8
KEYS:/NNC PROGRESS REPORT/

All,

Tonight the prototype NNC successfully interfaced to a Qume 142 Floppy Disk Drive and booted (and ran) Z-DOS under Z-CPR3.

I know this sounds pretty trivial, but I am excited!@

Lyle

C2974 CC124 Skip Hansen (WB6YMH,2964) 12/17/85 1Ø:48 PM L:24
KEYS:/PR-BS!/

Phil,
I usually stay away from the Datagram vrs VC wars, but your last comments on that old Gateway article hit a sour note!

1. The airborn "test" was on a unmodified VADCG board using unmodified VADCG software which was for a large part made operational by WA6JPR who received no credit in the various "PR" releases. The other side of the historical link was simply pluged into a modem which happened to be dialed into ARPA. BIG DEAL!

2. The 56K baud modem was shown at a ARRL convention in 1984, but none of the information was ever made available or published in any amateur publications even after numerious requests.

3. A large part of this work was done under research funding.

4. The work these guys do is great and does make us look like we are in the Stone age, but remember these are paid professionals who also happen to be Hams and are lucky enough to get paid to do what we do for fun.

Yea, I wish things were developing faster too, but unfortunately I work for a living where packet radio is not a company project. 73's Skip

C2974 CC125 Phil R. Karn (ka9q,2979) 12/18/85 1:38 AM L:48
KEYS:/SOMETIMES WHEN YOU REINVENT THE WHEEL YOU GET BACK A FLAT TIRE/

I'm sorry I hit a sour note, I knew very little more than what I read, plus a few mail conversations with the people involved. But to answer a few of your comments:

1. It would seem that the ability to run TCP/IP over even an unmodified VADCG board and to access all those services on the ARPANET without the other ends being aware that a new form of packet radio link was involved says a lot for the design of the ARPA protocols. I have yet to see an alternative approach that can claim the same flexibility.

If you can handle a variety of new jobs effectively with the tools you already have, I think I'd respect your toolmaker more than if his products were so specialized that you have to buy a whole new set for each new occasion. Unfortunately, toolmakers only make money by selling NEW tools, so...

2. If what you say about the modem is true, that's a real shame. We desperately need such a critter.

3. So what? Lots of things hams now have were first created under research funding by corporations or the government. Just because Zilog spent millions developing the Z-80 doesn't mean we can't benefit from it; they spent the money so we don't have to spend it ourselves. I guess it's the old "I built it all myself out of junkbox parts" syndrome, but of course the "REAL" hams build all their own capacitors and rectifiers too. Personally, I don't get much out of wheel (or protocol) reinvention. I prefer to spend my time building things I CAN'T buy off the shelf.

Sometimes just knowing that somebody has been able to do something is in itself very valuable. Even if you don't get the details, at least you know it's POSSIBLE. I know from my own work that much of a researcher's time is spent persuing dead ends. You're there because nobody else has been there before, and you never know if something will pan out until you try. If on the other hand you're trying to develop (or duplicate) an operational system out of existing technology, then doesn't it make sense to listen to people who have "been there" so you don't repeat their mistakes? You are certainly free to view their advice with a jaundiced eye and not follow it, but you shouldn't reject it out of hand just because of who funded it.

I keep sensing the widespread attitude that amateur packet radio is somehow totally unique in the world, and nothing that anyone did on the outside (most of all with government money) could possibly be relevant to our needs. This seems to be an awfully wasteful approach, especially when as you say, we aren't lucky enough to earn our livings off it.

Phil

C2974 CC126 Thomas A. Moulton (W2VY,995) 12/18/85 2:19 PM L:1Ø
KEYS:/CREDIT WHERE CREDIT IS DUE/
A: 125

A:12Ø
Howie, FYI RFC stands for Request For Comment, which means it's a Standard?

Phil,
In your last comment you praise ARPA for being able to plug in a VADCG board into an ARPA plug instead of the regular modem, I think you praise was sent the wrong way, after all these computers are connected via RS-232 interfaces and a packet board looks just like a TEL CO modem!

The basic function of the current packet boards is a point to point link

C2974 CC127 Phil R. Karn (ka9q,2979) 12/18/85 9:3Ø PM L:7Ø
KEYS:/PROTOCOLS/PROGRESS REPORT/

Howie, there is a timely and relevant paper in the December 1985 issue of IEEE Transactions on Communications: "Prevention of Store-and-Forward Deadlock in Computer Networks," by Inder S. Gopal. You might be interested. I can forward a paper copy if necessary.

Tom,
Even point-to-point modems vary widely in link speed, error rate and control interfaces. If the higher level protocols provide a very simple interface to the lower levels (e.g., link) and assume the minimum possible grade of service, you can use a waxed string and a pair of tin cans if it's capable of getting packets from here to there with > 0% probability.

All:
I am continuing work on my TCP/IP package. At this point, the pieces are

1. Low-level buffer primitives, utility functions, etc. (1.5 K on 8088) Essentially stable.

2. Transmission Control Protocol (TCP) module. (5.5K on 8088). Recently added support for Internet precedences (Routine, Priority, etc). Tested and working, no significant changes planned.

3. Internet Control Message Protocol (ICMP) module (1K on 8088). Recently written, under test. Common options are supported; some esoteric ones I haven't bothered with. Need to add proper Source Quench support.

4. User Datagram Protocol (UDP) module (750 bytes on 8088). Written, but untested. Needs redoing to provide more information to the user.

5. Internet Protocol (IP) module (1K on 8088). Supports sending and receiving IP datagrams on behalf of transport layer protocols. Recently added static table-based routing with primitives for adding and dropping entries from the routing table; makes source routing (complete path specification by datagram sender) unnecessary. No fully-automatic routing protocol is in yet; this will take more thought and work than I have at the moment. Haven't implemented full IP-level fragmentation/reassembly or IP option processing yet.

6. Serial Line IP (SLIP) module (400 bytes on 8088). Provides a convenient way to transfer IP datagrams across conventional asynchronous RS-232 interfaces, either hardwired lines or dialup modems. Tested and stable.

7. AX.25 link layer module (not rewritten yet). I am planning on writing a new AX.25 link layer for IP use, since I have come to the conclusion that the complexity of the complete LAPB protocol is unnecessary when just passing datagrams from point A to B. Each datagram will be sent in an AX.25 UI frame, and if link level acks are desired, the Poll bit will be set to request a one-for-one UA ack from the other end. To avoid the introduction of duplicates, I am planning on having a simple fixed-length FIFO list of datagram headers recently received by each IP module so that duplicates can be recognized and discarded without actually having to maintain lots of link level connection state.
This stripped-down link layer will also have to support the Address Resolution Protocol (ARP) for mapping IP addresses into AX.25 addresses without the need for a manually maintained lookup table.

Well, that's it from here. If anyone would like to play with this code, you're welcome to it, but unless you're serious about actually hacking on it I'd recommend that you wait until it's more complete so I don't have to send out too many updates. What's been written so far has compiled

and run in "self test" mode on about 4 different processor types without portability problems, although I'd like to find people willing to flush out any remaining portability problems with other C compilers.

73, Phil

C2974 CC128 Howard Goldstein (N2WX,2987) 12/18/85 9:33 PM L:2
KEYS:/UI/AND WHAT IS "SELF TEST" MODE?/
A: 127

No wonder

C2974 CC129 Howard Goldstein (N2WX,2987) 12/18/85 9:44 PM L:3
KEYS:/RFC/MAIL/
A: 126

Thanks Tom, I guess I should know that but you know how tough it is to remain well connected when all your mail gets lost (like they were datagrammes)

C2974 CC130 Phil R. Karn (ka9q,2979) 12/19/85 12:58 AM L:12
KEYS:/SELF TEST/ASK A VET IF HE LIKES VC'S/

Self-test mode (dubbed "incest-is-best mode" by one of my cohorts) is where you loop back the IP datagrams in software, create two TCP descriptors, establish a connection between the two and send some test data back and forth. It exercises just about everything in the TCP logic including the provisions for simultaneous connection establishment and closing, which almost never happen in real life.

I'd like to ask how many banks out there trust an X.25 network's assurances of reliability enough to do away completely with all forms of end-to-end checking on their $500,000,000 funds transfers...

Phil

C2974 CC131 Skip Hansen (WB6YMH,2964) 12/19/85 1:11 AM L:34
KEYS:/MORE PR BS/

Phil, I guess I shouldn't have started out my previous message by referencing the Datagram vrs VC wars. I only wanted to point out that the cited reference was in my opinion "PR hype" and that the actual experiment was not quite what one would assume from reading the Gateway (and other) writeups. I object to the comparision of the accomplishments (real or otherwise) of the Ham groups being compared with that of professional groups. I have nothing against them because of their TCP/IP leaning. The whole "test" just reminds me of when we got our first autopatchs going, we would impress the unknowing by calling information in the deep south (interesting sounding accents to Southern Californians) to ask what the weather was. The "acomplishment" was in getting a dial tone and being able to dialout, not in the distance covered by Bell after we dialed!

If the ISI guys had plugged their TNC into a different modem jack a X.25 circuit connection to Europe would have been just as impressive (or unimpressive!). TCP/IP was not used over the TNC link at that time so protocol flexibility was not tested.

As for not re-inventing the wheel, I am all for it when the previous wheel builders either publish the information on the design of the wheel or leave it around somewhere where it could "reverse engineered" It is hard to avoid anothers researchers mistakes when they are never published.

I am also specifically not saying that ISI hasn't developed some very worthwhile stuff after all, as I am sure you know, some of the ISI players were the "inventors" of ARPA net (their words not mine).

If "real hams" build their own capacitors maybe I am mistaken about ISI, they grow their own silicon!

And by the way of course I am a Trekie and proud of it. No offense taken. 73's Skip

C2974 CC132 Phil R. Karn (ka9q,2979) 12/19/85   3:17 AM L:24
(ORIG.)   12/19/85   3:13 AM L:22
KEYS:/AMATEUR PSYCHOLOGY/FOOD FOR THOUGHT/

Here's another thing to keep in mind while arguing the merits of different ways to build amateur packet radio networks.

The average amateur is very individualistic. It is much more difficult to get him to contribute to a joint effort resulting in a shared system than it is for him to buy another piece of hardware for his shack. AMSAT sees this all the time; the same person who gladly forks over $2000 for an HF KW linear he hardly needs balks at buying a $100 share of a communications satellite that will likely do far more for his communications capability. The linear he can park in his shack, twiddle the knobs and show off to his friends. The closest he ever gets to the satellite is a picture in the Amsat Satellite Journal; it isn't truly "real" to him.

What this says to me is that a network design philosophy that puts the lion's share of the "smarts" out in the "network" (as opposed to the amateur's shack) is going to rub against the grain of most amateurs; they won't support it. Oh, they'll all say they want it alright, until you ask them to kick in extra money to help support it. Given how much money most hams have invested in RF hardware just to avoid paying the telephone company, I suspect you'll find it easier to sell him a more expensive box than it'll be to sell him a cheap box that requires a bigger membership fee to talk beyond his own back yard.

Phil

C2974 CC133 Howard Goldstein (N2WX,2987) 12/19/85   8:15 AM L:12
KEYS:/QOS/IF IT COST $50 FOR AN OSCAR.../
A: 132

I don't see your point PHil, encryption and end-end reliability are, clearly, transport issues. The `flavor' of network running underneath is not germane (although the QOS might be).

I feel the argument that a VC network is too expensive is unfounded. There are hundreds of $50 Xerox 820s out there in (how shall I say this?) 'communal' amateur service.

Again, this point is rather unclear Phil since on the one hand you've claimed VC switches are more $$ to implement while on the other hand, you deplore the fact that IP embellishments, on-the-fly routing of each and every packet etc. are missing.

The very things you're touting would force the switch's cost up to beaucoup (sp) bucks!

C2974 CC134 Phil R. Karn (ka9q,2979) 12/19/85 12:32 PM L:7
KEYS:/HAM PSYCHOLOGY/

Note that I didn't mention "datagrams" or "virtual circuits" in that comment. I was only saying that if you have a choice between putting a certain function in the network or in the user's host, one would do well to consider the psychology of the average ham. Such functions might include reliability measures, protocol translations, etc.

Phil

C2974 CC135 Thomas A. Moulton (W2VY,995) 12/19/85 3:46 PM L:22
A: 134

I think that psychology is fading, after all see how many people want to have 2M repeaters up?

In any case I think that technically the more dispersed the network functions are the better off we will be, How many hams want to buy an MS-DOS system (aka 8088) for their Packet station?
Hell we have a hard enough time with getting them to buy anything other than a C64 or other Home computer (as to a Personal Computer)

And look at the people who got into packet with the software approach!

I think that we should remember that hams are cheap, lets get them in for a smaller box in their station.

after all don't the larger percentage of the USERS like to be part of the Club? (as in local club)

So!
I guess this still means,
You go your way, I'll go mine, I'm sure our packets will meet!

Happy Holidays ALL!

C2974 CC136 Phil R. Karn (ka9q,2979) 12/19/85 8:18 PM L:3
KEYS:/THOUGHT FOR THE DAY/

Those who believe that robust transport protocols are unnecessary because the network should be made "reliable" probably also believe that Star Wars will work.

M 24720 Gwyn Reedy (W1Bel,2975) 12/20/85 6:09 AM L:10
KEYS:/REQUEST FOR ARTICLES/
TO: (Group 95), W1Bel

ANNOUNCEMENT

The PSR Quarterly for January is undergoing construction at this time. There have been major inputs by WA7GXD and very little from anyone else. Please consider doing an article for the PSRQ during the beginning of the holiday period and get it to me by Dec 27th. Also I request you let me know that you are planning an input so that I may reserve some space for your work. Thanks.
Gwyn

C2974 CC137 Thomas A. Moulton (W2VY,995) 12/20/85 9:30 AM L:7
KEYS:/ROBUST/
A: 136

Phil,
I think we do BOTH have Robust protocols,

Robust: 1) strong and healthy, 4) full and rich
2) suited to or requiring physical strength or stamina
3) rough; coarse; boisterous

C2974 CC138 Phil R. Karn (ka9q,2979) 12/20/85 1:35 PM L:6
KEYS:/ROBUST/

I thought that in the context of a communication protocol, "robust" means that it's able to handle successfully a wide variety of unpredictable network behavior, including loss, misrouting, duplication, (often with large delays between the original copy and the duplicate) and misordering, without data being lost or erroneously delivered

to the user.

C2974 CC139 Thomas A. Moulton (W2VY,995) 12/20/85 9:43 PM L:6
KEYS:/ONLY TIME WILL TELL/YOU ARE ABOUT TO SEE AN X.400 EXPLOSION/
A: 138

Again, both are robust, TCP/IP and X.400/X.200
They will both work.
Each has it's technical advantages.
Each has it's draw backs

ar sk

C2974 CC140 Phil R. Karn (ka9q,2979) 12/21/85 2:47 AM L:12
KEYS:/THERE'LL BE AN EXPLOSION ALRIGHT/

I assume the explosion will occur when the users find out just what a bill of goods they've been sold. A "standard" that consists of five co-equal but mutually incompatible "classes" is no standard. Everybody but the European PTTs are already talking about punting everything but the TCP-lookalike TP-4. But if you're going to do that, why bother changing in the first place (unless the vendors see it as a chance to sell you a complete new set of tools...?)

Yes, the change is probably inevitable. So is the sun's red-giant phase. Which will occur first (or be more disruptive) is an open question.

Phil

M 25241 Howard Goldstein (N2WX,2987) 12/22/85 9:03 AM L:14
KEYS:/FLORIDA THRASHES SAREX2 SOFTWARE/
TO: (Group 95)

On the evening of Saturday, December 21, Henry WA4HXZ flew a TNC 2 with SAREX2 (release 1.1.2a), and a Motorola HT from South Florida up the middle of the state to near Lakeland and back down.

899 ROBOT QSOs were made, no real problems were reported, nor were any observed from my QTH.

However the combination of DIGIPEAT ON and FULLDUP ON was terrible. Whenever someone tried to digi thru it the TNC would kick on its transmitter right over acks, UAs, etc.

I suggest that the actual mission default ONE of these parameters to OFF. Comments?

73 Howie

PS I'd also make a recommendation that TNCs which send SABMs while connected should not be allowed, but we don't have to get into that...

C2974 CC141 Howard Goldstein (N2WX,2987) 12/21/85 9:20 PM L:7
KEYS:/ONLY $99 DOWN $99 A MONTH FOR 99 YEARS/
A: 140

I keep hearing that users are being horse traded into CCITT networks but there isn't any proof. I never hear that the people who sold them are on unemployment lines.

Where's the one iota to back up dem claims? How can a market survive, and grow, if customers are disatisfied and things are as bad as they're alledged...

C2974 CC142 Phil R. Karn (ka9q,2979) 12/22/85 3:26 AM L:33
(ORIG.) 12/22/85 2:53 AM L:25
KEYS:/CCITT/WITH FRIENDS LIKE THAT WHO NEEDS ENEMIES?/

I'm constantly reminded that there isn't necessarily a rational connection between the price of a service and the cost of providing it, but a look at the existing networks might be instructive.

If you look at Telenet's tariffs and figure how much it costs to ship files around, you will note that one of the new 9600 bps dialup modems plus good ol' AT&T will blow them away, any time of the day or night. At certain times much slower speeds (like 2400 baud) will break even. Uninet is even worse, since they charge by the character instead of by the packet.

All this is further support for my contention that the only thing an X.25 network is good for is remote terminal concentration. The problem is, the users are getting a lot more sophisticated than that. They want to interconnect their local area networks, which come in all shapes and sizes. Some are datagram (Ethernet CSMA/CD busses, Pronet token-passing rings), others are virtual circuit (your canonical "data PBX", although I don't think they deserve the term "local area network"). The CCITT is used to a world where they own everything, including the telephone on your desk, and they are uneasy with the idea of dealing with a proliferation of "customer premises networks" that they don't have any control over.

Since the CCITT has never worked in a heterogeneous world before, I have very little confidence in their ability to promulgate standards that work in such an environment. Certainly not when there's another organization (ARPA) which has already proven its ability to work in a true "internetwork" environment. No doubt the CCITT members would love to do away with all these "non-standard" LANS and protocols, and legislate X.25 right up to all your computers, and of course they'd be happy to rent you all the necessary equipment...

Phil

C2974 CC143 Howard Goldstein (N2WX,2987) 12/22/85 9:58 AM L:12
KEYS:/AND NOW/THE REST OF THE STORY/
A: 142

Yes comparing Telenet with AT&T is further proof of my point that datagramme networks are useful ONLY for transferring one file.

Be serious, no one has enough money to tie up an AT&T voice circuit all day. Telenet is the method of CHOICE for interconnecting many geographically dispersed LANS. Moreover, a large number of private networks (big stock brokers for instance) who have a real need for real systems use REAL CCITT protocols in the real world, and don't they excercise their choice over network "flavor"? They're apparantly paying real money and most are not overtly subsidized by the DoD/ARPA, or any other government agency.

Meaningless comparisons seem to abound when the datagramme crowd tries forcing analogies...

Good Day!

C2974 CC144 Phil R. Karn (ka9q,2979) 12/22/85 4:03 PM L:31
KEYS:/PDNS AND NETWORK INTERCONNECTION/

How is my Telenet vs AT&T comparison supposed to reflect badly on datagram networks? Telenet is internally virtual-circuit based, while AT&T is just a fixed bandwidth circuit switching network. My point was that Telenet and the other X.25 carriers are not as "cost effective" as many people believe, unless your bandwidth requirements average at very low values over long periods of time (i.e., represent interactive terminals). People interconnecting local area networks such as Ethernets want their computers to be able to talk to other computers,

and in this situation you need BOTH fast setup time and the ability to ship large files economically. Perhaps the picture will change as more of the PDNs install fiber backbones and competition brings prices down. However, the LAN-interconnect users will still be saddled with the hairy X.25 VC interface originally designed for slow speed terminals when they're trying to gateway their 10 megabit Ethernets together. Something different is desperately needed, and ISDN certainly isn't it.

I think you'll find from reading just about any paper on the subject of heterogeneous network interconnection that the datagram approach is the ONLY workable way. Xerox is a good example of this; we haven't talked about it much, but the Xerox Network Systems set of protocols had common beginnings with the ARPA Internet protocols, and despite differences in detail (address field formats, the distinction between the Internet and Transport layers, etc) they both rely on the notion of a standard "Internet Packet" datagram which is encapsulated in whatever link level protocol happens to be available.

I could cite plenty of other networks which were all built around the datagram concept, dispelling the erroneous notion that only the DoD likes them, but I'd probably bore you with the length of the list.

Phil

C2974 CC145 Phil R. Karn (ka9q,2979) 12/23/85   6:41 AM L:160
KEYS:/A MORE EFFICIENT LINK LEVEL PROTOCOL/COMMENTS WELCOME/

Recently I've been thinking about ways to improve the performance of link level protocols on noisy links, and I came across a very interesting idea.

In the paper "ALOHA Packet Broadcasting - A Retrospect", by Binder, et. al., published in the 1975 (10 years ago!) National Computer Conference, are the following interesting sections:

[KA9Q Note: the "MENEHUNE was the ALOHANET's packet switch, analogous to our TNC. Its name is an obscure wordplay. The ARPANET people were calling their packet switches "IMPs" (Interface Message Processors), so the ALOHA people simply chose the Hawaiian word for "imp", as in "mischevious dwarf".]

"Error control for broadcast channel data packets (MENEHUNE to user nodes) involves some special considerations. For efficient operation, the usual positive acknowledgment scheme in which the ACK's themselves are not acknowledged depends on a high probability of the ACK's being successfully received. However, an ACK sent from user nodes must compete with data traffic in the random access channel. At full channel loading each random access packet must be retransmitted an average of 1.7 times, which means each data packet or ACK must be sent a total of 2.7 times on the average before it is successfully received. But in order to force retransmission of the ACK's, the data packet being acknowledged must also be sent an average of 2.7 times by the MENEHUNE - even though it was received correctly the first time!....

[ka9q note: these numbers come from the fact that running a pure ALOHA channel at the optimum throughput level of 1/2e or 18.4% requires an "offered load" of 1/2 or 50%. I.e., the channel will have at LEAST one carrier on it 50% of the time, but only 18% of the time will there be ONLY one carrier. This means that each packet has a probability of (1/2e) / (1/2) = 1/e = 36.8% of making it across, and therefore packets must be sent on the average 1/(1/e) = e = 2.718 times to make it across once.]

"...to eliminate the unnecessary repetitions of data packets from the MENEHUNE, it is also necessary to acknowledge the ACK. That is, the ACK sent by a user node is timed out and retransmitted until an acknowledgement for it is received, just as for data packets. If another

packet is waiting for transmission to the node at this time, its transmission with the next sequence number constitutes the ACK to the ACK; otherwise, a short ACK-ACK packet is sent by the MENEHUNE. This can be easily shown to result in significantly less total channel overhead, at the expense of more complication in the node implementation."

I was intrigued by this idea and decided to study it further. What follows is my initial analysis of the ACK-ACK protocol and how it compares to the current AX.25 protocol.

In AX.25, ACKs are not acknowledged. This means that if the probability of getting a packet across a given channel is p, then on the average, A = 1/p transmissions of the ACK by the receiver will be necessary for the sender to get one back. To elicit 1/p ACKs from the receiver, the sender has to send D = 1/p * A = 1/(p**2) data packets. For example, if p = .5 (50% probability of getting a packet across the channel), then you can expect, on the average, A = 2 transmissions of the ACK and D = 4 transmissions of the data packet before sender can go on to the next data packet. If p = .25, then A = 4 and D = 16.

Note I've assumed for sake of illustration that p is equal in both directions; in the "real world" a data packet is a bigger target than an ACK and is therefore even more likely to get clobbered.

To show the effect of ACK-ACKs, I'll define a number N, the ratio between the sender's data-retransmission timer to the receiver's ACK-retransmission timer. We assume that this number is always greater than 1. (If it wasn't, the sender would always time out and resend the data before the receiver ever gets a chance to retransmit its ACK). N therefore represents the number of times the receiver will attempt to retransmit its ACK before the sender unnecessarily resends its data packet (which is what we'd like to avoid).

The probability P of getting an ACK through at least once in N transmission attempts when each attempt has probability p of succeeding is

P = 1 - (1 - p)**N

Make N large enough, and P approaches 1 (i.e., the ACK will almost certainly get through.) Of course there will still be plenty of data packet retransmissions, simply because (1-p) of them will never make it to the receiver in the first place. So the total expected number of data packet transmissions D changes from 1/(p**2) in the present case to (1/p) * (1/P) with ACK-ACKs. Let's plug in some numbers and see what effect this has on D, the expected number of data packet transmissions.

p = .5, N = 5, no ACK-ACK case:
    D = A * 1/p = 1/(p**2) = 1/(.5*2) = 4
same, with ACK-ACKs:
    P = .969
    D = (1/p) * (1/P) = 2.06, a 49% decrease

p = .25, N = 5, no ACK-ACKs:
    D = 16
same, with ACK-ACKs:
    P = .763
    D = 5.24, a 67% decrease

You may be asking "what about the extra overhead of the ACK-ACK"? Recall that a new data packet can serve as an ACK-ACK. So long as there is a steady stream of traffic, "standalone" ACK-ACKs are never transmitted; the next data packet provides that function. Only at the end of a traffic burst is an extra packet generated, and this seems a small price to pay to save all those unnecessary retransmissions.

So clearly we have something here. The question is how to implement it.

I am not certain that this technique can be adapted easily to LAPB, the connection-oriented upper sublayer of AX.25 Level 2. It may be possible, but it would take some work. LAPB was originally intended for highly reliable point-to-point links, and radical surgery may be necessary. In any event, LAPB is an unnecessarily complex link level protocol for a datagram network, so I already have a motivation for doing a lobotomy instead.

Fortunately, there are "hooks" in AX.25 that allow us to dispense with LAPB and implement the ACK-ACK protocol while still conforming to the AX.25 addressing format. We need three types of packets to implement the ACK-ACK protocol: a data packet, a regular ACK packet (sent in response to the data packet) and an ACK-ACK packet (sent in response to the regular acknowledgment when no additional data is available). Here's one possible assignment:

data packet - UI frame with or without Poll bit set
regular ack packet - UA, with or without Final bit set
ACK-ACK - empty UI frame with Poll bit cleared

The poll/final bits allow either or both acknowledgments to be optional. For example, stations in a datagram network operating on high quality links with very low packet loss rates might elect to operate without the overhead of link level acknowledgments; in this case, each data packet is sent in a UI frame with the poll bit cleared. Similarly, when regular acknowledgments are requested (i.e., with the poll bit set in UI data frames), the receiver uses the final bit in the UA to indicate whether an ACK-ACK is requested. Since a new data packet can be sent in place of an ACK-ACK, making both out of UI frames simplifies the receiver implementation. In each case, setting the poll/final bit means that some sort of response is expected from the other side, which also simplfies the implementation.

Notice that there are no sequence numbers, since a sliding window protocol such as LAPB is unnecessary on a half duplex channel. The protocol operates in a lock-step fashion; there is always a one-for-one relationship between a data packet and its acknowledgement, and this simplifies things enormously. Since packets cannot "cross in the mail" on a shared half-duplex channel, there is never any ambiguity as to which data packet an ACK refers to (note that this assumption is violated and the protocol breaks down in two situations: a long digipeater string where the end stations cannot hear each other directly, and on a long-delay satellite channel where the round trip delay exceeds the packet transmission time. Digipeaters we're already getting rid of, and I'll worry about high speed satellite channels when we get one.)

The only "state" required in the protocol is a buffer for the most recently sent or received packet. (The transmitter may need to resend it and the receiver needs to detect and filter out duplicates transmissions). However, once a packet has been sent and fully acknowledged, no state needs to be maintained, a MAJOR advantage when servicing a large number of mostly idle local "links" with limited memory.

Comments and suggestions on this idea are welcome.

73,

Phil Karn, KA9Q

C2974 CC146 Jack Brindle (WA4FIB,2963) 12/27/85 12:42 AM L:73
KEYS:/LAYER 3 AND 4 BOTH NEEDED/PERHAPS TCP/IP HAS SOME GOOD IDEAS!/

I guess I will now jump into the discussion. Howie and I have been having a "private" discussion on the layer 3 - 4 issues, which should be carried on here.

First, we should all realize that ISO and CCITT are two completely different organizations with seperate goals and purposes. They just happened to have adopted similar communications models.

The first thing we must take care of in our system is to assure the

By the way, Stoner's proposal for the Public Digital Radio Service

delivery of data from one end user to another. This is accomplished in the OSI model with the transport layer. The network layer is incapable of doing this function since it does not "see" the information that is needed to perform the function. The network layer simply guides the data through the network from node to node. If you don't believe this then it is time to look at the information supplied by both ISO & CCITT.

Because of this, any implimentation of a network layer without the transport layer will suffer lots of problems. An example of this on a smaller level is the current state of file transfers on packet. File transfers are generally not done in packet radio (RLI BBS transfers are really message exchanges - look at the problems had in long file exchanges). The reason is the lack of a decent transport mechanism above the link layer. The link layer can "guarantee" the delivery of a small number of packets, but it cannot "guarantee" they will all arrive, and get there in order. Attempts at using XMODEM have tried to overcome this problem, but have met with other difficulties.

This is not an attempt to "advertise" my session layer protocol, but rather points out that node-to-node error checking is not good enough for the file transfers than will be required in the future. I maintain that packet radio in the future will be quite different than it is now, and must support far more and better services than we are willing to support now. The place for error checking is in the end user's systems. This is done at layer 4. Now if anyone wishes to release a layer 3 coding with a layer 4, be aware that ridicule will come. (But then it has come to those who wrote the layer 2 spec we all use...)

Probably the biggest advantage that the TCP/IP folks have to offer is that they have a "well-designed" system. Read that again - they have a system. One that was designed for all the pieces to fit together. One of the biggest problems in the packet world today is the lack of CCITT standards above layer 3 (at least until recently). Although packet switches work well at switching packets, it is extremely difficult to get terminal controllers from different manufacturers to work together. This is because each vendor used a different spec for layers 4, 5, 6 and 7. Not even the ISO and CCITT agree on the exact purpose of layer 7. (CCITT puts its actual drivers and hardware (read screen drivers) interface code here, whi while the ISO uses it as an interface inside a computer). This problem also exists in ham packet radio. We don't have a system design. We are building the system up from the inside out. What results will invariably become a kludge. The thing that really bothers me is that this "kludge" is quickly being adopted by the low-end players of the commercial world!

Now I propose that the arguing over end-to-end acking versus node-to-node be placed aside. The answer, of course, is that we really need both. Instead why don't we draft a layer 4 spec and get started on working to design and code it.

Of course, we could follow Phil's lead and adopt TCP/IP with all of its ramifications including a complete system design...

Phil; I am indeed interested in running your code. I have just finished up MacPacket/TNC2term (and shipped my backlog last Monday, the 23rd), and will shortly be moving from temporary quarters into a house. I will be able to devote time to integrating the code into my MacPacket package for demonstration purposes. I managed to lose your address, please give it again. This time I promise to get a large envelope to you (with suitable postage, whatever that is) so that I can get a diskette and whatever docs you would like to send.

I again also extend the offer to the VC folks. Give me a good layer 3 & 4 implimentation and I will code it into MacPacket also. I would love to be able to run this stuff at Orlando to help out in our discussions.

Another note: Those of you on Compuserve, take a look at the files FCC1.DOC and FCC2.DOC in Public Access. (GO ACCESS). They contain a very interesting document from a ham in Calif that is proposing a Public Data Service to run on a portion of the "unused" amateir 6 meter band at up to 1 MB/s! I'm not sure if this guy is just dreaming, but the FCC has decided to take some sort of action on his petition. This looks like something that we need to channel back into ham radio. I'm not a supporter of "no-code", but perhaps it is an idea whose time has come!?

Good night all - I hope everyone had a Merry Christmas.

C2974 CC147 Phil R. Karn (ka9q,2979) 12/28/85   3:34 AM L:53
KEYS:/NOW THAT WE HAVE LEVELS 3 AND 4 LETS MOVE ON/

Jack, thank you very much for your comments. I am happy to hear them; I've been saying similar things for over a year now!

My TCP, as a "layer 4" protocol, is up and running. It implements the full-blown ARPA TCP spec as described in RFC-793 and MIL-STD-1778 with two exceptions: security (which I assume we have no use for) and urgent data (which I can't figure out how to use, although it wouldn't be hard to add). Other than that, there was no reason to change the spec to suit our needs, with the result that my implementation is fully compatible (and has been tested with) the various other implementations out on the Internet. I hereby toss it out on the table for consideration as our end-to-end Transport protocol. There is no reason to "roll our own", thus adding to the layer 4-5-6-7 confusion you alluded to. In response to comments regarding the programming interface, I've reworked it to make the facility more general; there are now optional "upcalls" (also known as "pseudo interrupts" for transmit buffer availability and connection state change as well as receive data availability.

As I mentioned earlier, the ARPA Internet has no need for a distinct session layer (level 5); its few functions are assumed by the "well known port" concept in TCP (and UDP). As for presentation layers, this depends on the application. Since remote login is likely to be an important initial application, I'd like to suggest that we consider the ARPA TELNET protocol. This is a simple, extensible technique where various options (e.g., which end of a connection is to perform character echoing) can be negotiated. For example, a user connecting into a UNIX system (which conventionally operates in "full duplex" mode with per-character transmissions and remote echoing, to allow use of fancy command line and screen text editors) will receive a WILL ECHO offer from the host. Your Telnet, which is under your control, can respond with either a DO ECHO or a DON'T ECHO command, which either accepts or rejects the remote host's offer. Many other options exist, including ones to control the faucets on the kitchen sink, but only a small set (about 5) are actually recommended for use and only one or two (echo being the most important) are actually used widely. The mechanism is there, however, for other options to be implemented, and their use can be negotiated with other systems as needed.

Other presentation level protocols of interest include the FTP (File Transfer Protocol) and the SMTP (Simple Mail Transfer Protocol) with self-evident uses. SMTP really is simple to implement, as there are only about 5 commands. FTP is quite a bit more complicated, since it is one of the oldest commands in the ARPANET (long predating TCP/IP and about as old as TELNET), and although it is widespread there may be simpler alternatives. I would like people to look at SMTP, however, since it represents a much "cleaner" way to transfer mail messages than the current highly ad-hoc of linked WØRLI bulletin boards. I can provide documents on these protocols, along with sample sessions, if there is interest.

Phil

C2974 CC148 Scott Loftesness (W3VS,2961) 12/29/85   3:27 AM L:1Ø
KEYS:/BRINDLE GET WITH IT/

Hey, Brindle, when are you going to share the specifications on those new protocols of yours? I thought you told us over a month ago that they're be here Real Soon Now! Did I miss 'em?

By the way, Stoner's proposal for the Public Digital Radio Service sounds good to me. Hell, implementation of that service and the definition of the protocol should keep you protocol warriors fully employed for years!

[Flame off!]

M 27278 Howard Goldstein (N2WX,2987) 12/31/85   2:59 PM L:41
KEYS:/AND BEST WISHES FOR A PEACEFUL AND PROSPEROUS NEW YEAR/
TO: (Group 95)

to: All
fm: You guessed it, Howie
re: It's 10PM, do you know where your vaporware is?
dist: Wide and far

The local digipeater (N2WX-7 a.k.a MLB) is now also a network node. I am not sure if it is the first instance of ax.25 13 AND the stuff to do the pad functions for folks unlucky enough not to have a TNC 2, but I include the example below of a short session with the Orlando mailbox using the PAD.

Networking, and ax.25 level 3 are REAL. They're HERE. ANYONE can use them.

Of course this will all be available for anyone's use from the Orlando conference.

```
actual unretouched example of the WORKING ON THE AIR pad:
:\
cmd:c n2wx-7
cmd:*** CONNECTED to N2WX-7
{4
gator 2 pad 03100305724 port A

enter: call [,digi1 [,digi2 [,digi3] ] ]
to?k4aho orl
*** pad: connection reset
K4AHO Packet BBS, Orlando, FL, new msg's 1126 to 1125
Type X for long menu, H for Help
Last Msg 1125, 59 Active Msgs (B,D,G,H,I,J,L,R,S,T,U,W,X) >
Oh, Hello Howie
Last Msg 1125, 59 Active Msgs (B,D,G,H,I,J,L,R,S,T,U,W,X) >
ll 3
Msg# TR  Size  To      From    @ BBS  Date     Title
1125  N   248 WD4KAV  WB4TRJ          851231 greetings
1120  N   208 ALL     WD4IXD          851230 voice net pt.2
1119  Y   257 K4AHO   N4LXI           851230 BBS
Last Msg 1125, 59 Active Msgs (B,D,G,H,I,J,L,R,S,T,U,W,X) >
b
*** DISCONNECTED
N2WX-7>K4AHO,ORL <UA>
N2WX-7>N2WX-1 <D>
```

M 27337 J. Gordon Beattie Jr. (N2DSY,2990) 12/31/85   8:33 PM L:29
KEYS:/WE DID IT IN '85 !/
TO: (Group 95)

I have been a bit quiet for the last few months frankly because I hate going around in circles, but I will make a comment on the end of the year happenings.
(Pardon the condescension, but some of the comments have been frustrating.)

We have gone a year and we finally have a few working implementations of level 3 network components using the OSI and ARPA models.
THIS IS GOOD !

Regarding requests for protocol specs: we should stick to readily available

M 27278 Howard Goldstein (N2WX,2987) 12/31/85 2:59 PM L:41
KEYS:/AND BEST WISHES FOR A PEACEFUL AND PROSPEROUS NEW YEAR/
TO: (Group 95)

to: All
fm: You guessed it, Howie
re: It's 1ØPM, do you know where your vaporware is?
dist: Wide and far

The local digipeater (N2WX-7 a.k.a MLB) is now also a network node. I am not sure if it is the first instance of ax.25 l3 AND the stuff to do the pad functions for folks unlucky enough not to have a TNC 2, but I include the example below of a short session with the Orlando mailbox using the PAD.

Networking, and ax.25 level 3 are REAL. They're HERE. ANYONE can use them.

Of course this will all be available for anyone's use from the Orlando conference.

----

actual unretouched example of the WORKING ON THE AIR pad:

```
:\
cmd:c n2wx-7
cmd:*** CONNECTED to N2WX-7
{4
gator 2 pad Ø31ØØ3Ø5724 port A

enter: call [,digi1 [,digi2 [,digi3] ] ]
to?k4aho orl
*** pad: connection reset
K4AHO Packet BBS, Orlando, FL, new msg's 1126 to 1125
Type X for long menu, H for Help
Last Msg 1125, 59 Active Msgs (B,D,G,H,I,J,L,R,S,T,U,W,X) >
Oh, Hello Howie
Last Msg 1125, 59 Active Msgs (B,D,G,H,I,J,L,R,S,T,U,W,X) >
ll 3
Msg# TR  Size  To     From    @ BBS  Date    Title
1125  N   248 WD4KAV WB4TRJ          851231 greetings
112Ø  N   2Ø8 ALL    WD4IXD          85123Ø voice net pt.2
1119  Y   257 K4AHO  N4LXI           85123Ø BBS
Last Msg 1125, 59 Active Msgs (B,D,G,H,I,J,L,R,S,T,U,W,X) >
b
*** DISCONNECTED
N2WX-7>K4AHO,ORL <UA>
N2WX-7>N2WX-1 <D>
```

M 27337 J. Gordon Beattie Jr. (N2DSY,299Ø) 12/31/85 8:33 PM L:29
KEYS:/WE DID IT IN '85 !/
TO: (Group 95)

I have been a bit quiet for the last few months frankly because I hate going around in circles, but I will make a comment on the end of the year happenings.
(Pardon the condescension, but some of the comments have been frustrating.)

We have gone a year and we finally have a few working implementations of level 3 network components using the OSI and ARPA models.
THIS IS GOOD !

Regarding requests for protocol specs: we should stick to readily available standards in order that we are able to scoff what we can in the way of experience, software and hardware. I am a bit concerned that Jack's work is undocumented, but I am more concerned that it is an unknown protocol with actual work done on an implementation. Vacuums are a horrible place to live or work !
The spec is important Jack, but the work your doing with the terminal programmes

needs to have something that others can talk to.

Last but NOT LEAST: THANKS TO LYLE, PETE, AND HOWIE
for their work on the TNC-2 and NNC.
WE HAVE LEVEL 3 VIRTUAL CIRCUIT NETWORKING IN 1985 !
Now WE must go and install the networks !
Have a GOOD YEAR !

73, Gordon, N2DSY

P.S. C U in Orlando !

M 27386 Pete Eaton (WB9FLW,2970) 1/ 1/86 1:47 AM L:11
KEYS:/PAT YOURSELVES ON THE BACK-NOW FOR A SIMPLE PROJECT/
TO: (Group 95)

Gordon:

Thanks for the kind words on TNC-2. Everyone on DRNET, and many others who are not, also sweated the last year out. So everybody pat yourself on the back!

Ok, now for a "simple project", anybody want to build a RF deck?

Happy New Year!

Pete

M 27613 GARY GARRIOTT (VITA,568) 1/ 2/86 8:48 AM L:6
KEYS:/TNC-2 MATING/
TO: (Group 95), VITA

DOES ANYBODY OUT THERE KNOW OF ANY MATING PROBLEMS OF THE TNC-2 WITH RADIO SHACK MODEL 100 OR 200? HAVE HEARD RUMOR TO THIS EFFECT... WE'D LIKE TO TAKE EITHER THE 100 OR 200 TO ETHIOPIA...

GARY

M 27638 Thomas A. Moulton (W2VY,995) 1/ 2/86 10:02 AM L:3
KEYS:/TNC-2 MATING/
A: 27613 TO: (Group 95), VITA

One problem I have seen is that when using the built in terminal program the M 100 seems to send X-OFF and never sends an X-ON to allow the tnc to send to the M100 again

M 27809 Pete Eaton (WB9FLW,2970) 1/ 2/86 7:43 PM L:13
KEYS:/THIS IS GOING TO BE HARD TO BELIEVE/
TO: (Group 95)

To: All
From: Pete WB9FLW
Subject: RLI Mailbox/Gateway On-Line
Dist: Open

I know this is going to be hard to believe but as of 4:30 P.M. Central Time today I finally got the RLI system back up including the 20 meter Gateway. I'll leave the system on 24 Hrs aday, be looking forward to chatting with you folks on the "other" network!

C2974 CC150 Phil R. Karn (ka9q,2979) 1/ 4/86 12:04 AM L:19
KEYS:/NETWORK PRE-EMPTION/

As I mentioned in my writeup on the Internet protocols, I support the notion of packet precedences or priorities. It is possible to queue packets for transmission based on these priorities:

for example, a node would transmit all packets marked PRIORITY before transmitting any packets marked WELFARE, and so on.

It is also possible for a node to preempt (i.e., abort) the transmission of a packet should a higher precedence packet need the same link. There are arguments each way; clearly preemption will grant the lowest possible delay to urgent traffic when the links are already busy with routine traffic, but at the expense of lower overall efficiency (because of all the aborted packets). I am looking for suggestions on the best way to handle this. I could preempt whenever a higher priority packet needs the link; I could preempt only for the highest priority (EMERGENCY), or I could preempt if the relative priorities differ by more than 1 level. Comments, especially from traffic handlers, would be appreciated.

73, Phil

C2974 CC151 Lyle Johnson (WA7GXD,2973) 1/ 4/86 12:36 AM L:36
KEYS:/NNC PROGRESS REPORT/

To: Anyone Interested
Fm: TAPR
Re: NNC Prototype debugging progress
(Ø3 jan 1986)

The NNC prototype testing is progressing fairly smoothly. The test unit is chugging away at a system clock of 4.6Ø8 MHz (due to CMOS Z8Ø"A" peripherals being available, while "B:" parts are not).

Using test code provided by N2WX, I have verified the operation of one channel of 1 SIO running HDLC data in a digital loopback mode at a data rate of 76.8 kbps (it can't hack 153.6 kbps in fulldup with the test code, however...). This is polling, not interrupt-driven. I have also verified that I can talk to both channels of both SIO/2s.

Similarly, I have verified that I can talk to both channels of the PIO and ran a simple test program tonight to drive a "Centronics" (well, OkiData) printer.

I have also written data to and read data from the SCSI chip (538Ø) but it is basically gibberish, I haven't attempted to make it do anything useful.

Finally, I have patched the BIOS to support the 4.6Ø8 Mhz clock rate and all the software I have written/used is running on the NNC prototype under the Z-system.

I have a pair of Mitsubishi drives on qualifaction loan, and Fujitsu and Tandon drives are enroute for the same purpose. The Mitsubishi drives are quiet and fast...

Oh yes, the prototype is using 8k-byte SRAMs - NEC didn't accept my offer to purchase 32k-byte parts at prices I was willing to pay! (but, I did get 1 free sample!!!).

More later. Lyle

C2987 CC246 Lyle Johnson (WA7GXD,2973) 1/ 5/86 12:54 AM L:1Ø
KEYS:/LOTSA RAM INNA CHIP/WORKS FINE/

I have successfully interface a 32k byte SRAM to a tnc 2 rev 2 board. Simply remove bb power (as well as regular power!), remove the two 6264ls from u24 and u25, and install the 43256 SRAM at u25. reconnect the bbram link, power up, initialize and go for it.

Oh yes, the 43256s cost only about $1ØØ each in small quantities...

Howie, got any code that can make use of 32k bytes of bbRAM?

Lyle

M 29254 GARY GARRIOTT (VITA,568)   1/ 7/86   7:48 AM L:21
KEYS:/URGENT FEEDBACK NEEDED/
TO: (Group 95), VITA

YESTERDAY I HAD A CHAT WITH DON MERTEN (K2AAA) WHO WAS REFERRED TO ME BY ALEX MONTARE, KA2VLP. DON HAS WORKED FOR MANY YEARS WITH DIGITAL TRANSMISSION METHODS OVERSEAS AND INSISTS THAT AMTOR, RATHER THAN PACKET, IS THE PREFERRED MODE FOR HF. ACCORDING TO DON, THIS IS SO BECAUSE FREQUENCY STABILITY IS MUCH MORE DIFFICULT ON HF PLUS THE RELIABILITY FACTOR, IE, FEWER BYTES PER BURST ON AMTOR TRANSLATE INTO MUCH HIGHER EFFICIENCY GIVEN STATIC CRASHES, ETC. DON IS COMING TO TOWN TODAY AND I MAY SEE AN AMTOR DEMO TONIGHT.

NATURALLY, THIS IS SOMEWHAT DISTURBING SINCE WE HAD ALREADY PURCHASED THREE TNC 2'S FOR THE ETHIOPIA PROJECT, BUT DON HAS ME JUST ABOUT CONVINCED THAT WE BETTER GO OVER THERE WITH AN AMTOR CAPABILITY AS WELL AS PACKET WHICH MEANS THE AEA "PACK-RAT" (PK-64)/COMMODORE 64 COMBINATION.

GUYS, IF WE GO TO ETHIOPIA AND IT DOESN'T WORK, OUR (I SHOULD SAY MY) GOOSE IS COOKED. I REALIZE THERE ISN'T MUCH COLLECTIVE EXPERIENCE OF PACKET ON HF IN THIS GROUP, BUT WOULD APPRECIATE SOME FEEDBACK ON THIS NEW DEVELOPMENT.

COULD DON BE RIGHT?

GARY

M 29258 J. Gordon Beattie Jr. (N2DSY,2990)   1/ 7/86   8:15 AM L:16
KEYS:/YOUR GOOSE IS NOT COOKED/
TO: VITA, (Group 95)

Gary,
There are those who can make a case for AMTOR/SITOR, but there is more to the story:

packet will NEVER allow a character error of any kind - AMTOR will.
there are bunches of packet stations of 14.103 that ARE working fine.
There is a 10 MHz frequency also and I can't recall it.
Packet uses a Timeout on error then retransmit philosophy, AMTOR/SITOR uses a FEC technique. Packet uses a multipacket rotating window acknowledgement system and AMTOR/SITOR uses a basic ack for each data system.
Most HF packeteers keep their packet lengths short so that the packets are less susceptable to hits during transmission. This takes care of the reliability question.
73, Gordon N2DSY

M 29322 Thomas A. Moulton (W2VY,995)   1/ 7/86 12:42 PM L:9
KEYS:/URGENT FEEDBACK NEEDED/
A: 29254   TO: (Group 95), VITA

I think that Don is assuming that you will use 128 byte packets, with packets that long he is right.

I don't know what the guys on HF mail forwarding use, but I think you could use packet sizes of 64 or 32 bytes long.

I have sent many messages that have gone via HF, it DOES work!
I realize that it is about twice as far to ET
Lets see what the rest of the group has to say.

M 29692 GARY GARRIOTT (VITA,568)   1/ 8/86   7:33 AM L:10
KEYS:/THANX FOR FEEDBACK/

KEYS:/THANX FOR FEEDBACK/
TO: (Group 95), VITA

THANX EVERYONE FOR THE COMMENTS...I'VE NOW SEEN DON'S SET-UP AND IT IS IMPRESSIVE...THE FACT THAT HE HELPED INSTALL THE VERY RADIOS IN ETHIOPIA SOME YEARS AGO IS ANOTHER FACT...HE DOESN'T THINK TIMING WILL BE A PROBLEM VIS-AVIS THE X-R RELAY....AM LEANING TOWARD HAVING A DUAL DEMO CAPABILITY JUST TO BE ON THE SAFE SIDE...

FURTHER THOUGHTS WELCOME...

GARY

M 29707 Paul Newland (ad7i,2978)   1/ 8/86   8:39 AM L:53
KEYS:/AMTOR/PACKET/HF/
TO: (Group 95)

FROM: paul newland, ad7i
TO: gary garriott
SUBJ: packet/amtor on hf
DIST: limited
DATE: 86.01.08

gary -- allow me to add my comments to those already made.
I hope that it doesn't further confuse the issue.
I have a lot of HF amtor experience but NO packet experience
at HF. So, I come with some strong biases.
Statements have been made that packet never allows an error
to get through. I would recommend that the author of such
statements rephrase that to be less misleading. I would suggest
that packet offers extremely good error control, but not perfect.
All single bit errors will be detected but there is a non-zero
probability that a packet with multiple errors might be accepted
as correct. However, the probability is remote (~1E-5).
The MODE-A operation of amtor alluded to is certainly not Forward
Error Control (FEC). It is an ACK/NAK as opposed to the ACK/NOTHING
protocol of packet. one strong advantage of packet here is that
the ACKs themselves of packet are much better protected against
errors than the ACKs of amtor.

Several folks have attempted to use fairly unsophisticated
analytic methods to compare amtor and packet. They have resulted
in unsophisticated results. I woundn't place too much faith
in such results. I can tell you this however, both work and work
very well. I will speculate (and I underscore the word "speculate")
that amtor will work better in a slugfest on 80 or 40 meters when
packet won't get through, because the channel error rate is 1E-2.
The price you pay such poor channel performance is no data throughput
with packet and a low throughput with poorer error probability
for amtor. Look at Peter Martinez's article
in QST 1981, pp25-27. If I remember right, he gives some throughput
and error probability numbers for different channel error rates
for amtor.
I don't recall your exact application but if you want to move a
large amount of computer data at 10 MHz or higher and can wait
for up to 24 hours (i.e., wait for best propagation of the day),
and you insist on an error rate of 1E-9,
then packet is probably the best way to go. It's tough to
argue with success; packet works every day on 10 and 14 MHz.

If you want to move a smaller amount of data (i.e., 3 pages or less)
and want to do at on 80 meters and do it right now and you
can live with an error rate of 1E-5, then AMTOR will suit your
needs best.
Bottom line, both work fine! I doubt that your goose will even
need to wiggle its wings, let alone spending time in the pot.

73, paul

M 29768 J. Gordon Beattie Jr. (N2DSY,2990) 1/ 8/86 12:24 PM L:9
KEYS:/AMTOR/PACKET/HF/OOOOOPS !/
TO: (Group 95)

I agree with Paul, but I'm not sure if I'd want 1E-5 when I could have 1E-9.
Sometime s promptness is more important though...
In reference to the term "errors", I would like to note that when a link breaks down (loss, resets, etc.) I want an indication. AX.25 does provide that to the user. I don't know what AMTOR does, but I would assume it has a recovery mechanism.
Pardon the faux pas Paul.
73, Gordon

M 11 Scott Loftesness (W3VS,2961) 1/ 9/86 1:12 AM L:119
KEYS:/HAPN PC-PACKET BOARD/
TO: (Group 95)

Msg#	TR	Size	To	From	@ BBS	Date	Title
231	BY	5291	ALL	AA4RE		860106	IBM PC based TNC announced

```
*************************************************************************
**                                                                     **
**            Hamilton Area Packet Network introduces the              **
**                                                                     **
**                     HAPN PC-PACKET ADAPTER                          **
**                                                                     **
*************************************************************************
```

The HAPN adapter is unique in terms of its versatility and ease of operation. It is designed for the IBM PC and compatibles. The IBM PC or compatible is rapidly becoming a leader in the HAM-world due its attractive price and excellent performance.

The HAPN packet adapter is a card that plugs directly into one of the slots of the PC. It contains a built-in modem which interfaces to the radio with a DB9 connector. The card is 8.5 inches (22 cm) long and contains a prototype area. The software is powerful and is a delight to use. The transfer of ascii or binary files is very simple by using the function keys on the keyboard. A popup menu allows you to set up the repeater routing table. Optional software is available for bulletin board operation. Experimental protocols such as VADCG V1 and V2 are also available.

Features of the H.A.P.N. packet adapter :

1. In addition to the American AX.25 protocol it also runs the Canadian experimental protocols V1, V2 as designed by Doug Lockhart (VE7APU) of the VADCG (Vancouver Amateur Digital Communications Group). (Doug is a world leader in modern packet radio since 1979 and founder of the VADCG.)

2. Excellent software. The software provided by HAPN is user friendly, easy to operate and very powerful. Available at present:
   a. AX.25 host support (included)
   b. bulletin board program PCRBBS (optional)
   c. file transfer programs (optional)
   d. experimental protocol support for VADCG V1, V2 (optional)

3. The HAPN adapter plugs into a PC, PC-XT, PC-AT or compatible. No external cabinet or power supply is required.

4. Simple to interface to your radio. It requires RX audio, TX audio and push-to-talk. The carrier detect (radio channel busy) is generated internally from the received data or optionally brought out from your

internally from the received data or optionally brought out from your squelch circuit.

5. The built-in modem is Bell 202 compatible. It uses the XR2206 and XR2211 chipset. The tones are 1200 and 2200hz. The data rate is 1200 baud. A hardware watchdog circuit prevents possible TX runaway problems as a result of power glitches or lightning.

   NOTE: The baudrate for the adapter is selectable by jumper from 75 baud to 9600 baud. (The on-board modem supports up to 1200 baud.)

6. The board contains a prototyping area for personal use. The area is pre-drilled.

7. The adapter is based on the INTEL 8273 HDLC/SDLC protocol controller chip with a straightforward interface to the PC.

8. The low level software (device driver) is interrupt driven and stays resident in the PC allowing the adapter to function in the background. An application program such as bulletin board, file transfer or terminal program does not have to be resident for the adapter to function. In this mode, the user is alerted to a caller by a connect alarm utilizing the PC's speaker.

9. Easy customizing for the user. The user configuration such as callsign/SSID, auto logging of packet activity to disk, default repeater, connect alarm etc. are set up using the configuration screen and remembered by the program. These options can be changed at any time.

10. NOTE FOR PROGRAMMERS
    Since the adapter uses the native code of the host computer no cross compilers are needed for anyone wanting to develop his own code.

ORDERING INFORMATION AND PRICE :

All prices are in US$ and include postage and handling in USA or Canada. Please add 3$ for overseas orders.

1.	Assembled and tested board with AX.25 host software and self-test program (4 programs and documentation)	$199.00
2.	Bare board with AX.25 software and construction details	$ 75.00
3.	AX.25 software only	$ 40.00
4.	Bulletin board & file transfer programs	$ 25.00
5.	Advanced VADCG experimental software	$ 25.00

Make checks and money orders payable to H.A.P.N. The address is :

HAPN
Box 4466, Station D,
Hamilton, Ontario,
Canada, L8V4S7

A Note on HAPN

The HAPN (Hamilton Area Packet Network) is a Packet Radio club founded in 1980 and dedicated to furthering the state of packet radio. It is currently involved in the design of a station node controller. The station node controller is capable of unifying a packet radio network by providing message routing, gateways and high-speed links with other nodes. Money raised with projects like this helps us pay for research

and development.

HAPN is also involved in experimentation with 4800 baud modems, and experimentation with the newly developed VADCG protocol V3.

[Via W3VS]

C2974 CC153 Harold Price (NK6K,2972)   1/ 6/86   3:59 AM L:393
KEYS:/LINKING SCHEME FORWARDED FROM COMPUSERVE/

Don't think I saw this on DRNET. It is forwarded from COMPUSERVE.
Item 23 entered Fri, Nov 22, 1985 (21:06) by Michael McClary
Proposal for a geometric organization of a ham long-haul network.

Things I've seen in previous items give me the impression that Michigan hams, at least, are thinking of hooking up the long-haul portions of a packet network with radio links pointed toward each of the four compass points, using the four ports on a Network Node Controller from TAPR or the like. This strikes me as being a bad fit to the real world, and I have a counterproposal. (It makes very good use of the NNC's four ports, by the way.)

If a ham long-haul network is set up, it will likely have to consist of network nodes which connect to high-speed long-distance links, plus a local area network (to get the individual hams' data on and off the long-haul links). Since it already is an expensive, multi-antenna and multi-radio box, it would be reasonable to place it in a location central to a large community, and make its LAN link strong and sensitive, making its coverage look much like a digipeater with the ability to "autopatch" into the long-haul network.

Each NNC's network-access LAN will be directly (i.e., no digipeater hops necessary) accessible from an essentially circular service area. In order to provide high-quality (no digipeaters between you and the NNC) service to as many hams as possible, these circles should cover the entire land area serviced. In order to avoid wasting resources, they should overlap as little as possible. This packing of circles is a hexagonal (honeycomb) array, which tiles the plane quite nicely. (Failing this, each should serve an area that has many hams, and the dead spots should fall where hams are few.)

The hexagonal array has a hidden benefit. The economic forces which governed the placement of cities, towns, and villages caused them to be laid out in a hexagonal array as well. (Actually, they form nested hexagonal arrays on varying scales - a few big cities in an array with large cells, thousands of villages on an array with tiny ones, and some of the village sites occupied by cities.) (You may not notice the hexagonal layout on a map because it is slightly distorted. The towns are spaced for equal travel difficulty, rather than equal distance, so they are a little closer together if separated by bad terrain.) As a result of this layout, you have cities right where you want to put your NNCs.

At first glance, it looks like each NNC will have 6 nearest neighbors, and thus will need 6 high-speed links. It turns out you can get away with three. Consider the following array:

```
^^

      /   \   /   \
    v    v    v
    !1   !2   !
    ^   (A)    ^
  /   \   /   \   /   \
vvvv
!   6!   Ø!   3!
^      (C)     (B)^
  \   /   \   /   \   /
```

... They are

```
    v    v    v
    !5   !4   !
    ^    ^    ^

      \   /   \   /
vv
```

Suppose the numbered sites are major cities. Zero has 6 nearest neighbor cities. To talk to them directly would require 6 links, and each link would have to be able to carry over a distance sqrt(3) times as long as the reach of the LAN's transceiver. The intersections marked with "v", "^", "(A)", "(B)", and "(C)", however, would normally contain small towns, and would be just on the edge of the NNC's LAN range. Suppose each of these put up their own NNC, also with three long-distance links, but not necessarily with a LAN of its own. (Everyone in the town should be able to reach the LAN of one or more neighboring cities.) "Ø" would now talk to "A", "B", and "C", "A" would talk to "Ø", "1", and "2", and so on.

This sort of network can be built with four-port NNCs, and with four transceivers at the major nodes, and three at the minor nodes. It doesn't require extra-long range transceivers for the inter-NNC hops. (The directional antenna will give enough extra gain to make the inter-NNC link solid, even though it might be carrying a higher data rate.) REALLY large cities (which tend to form another, larger hexagonal structure superimposed on this one) can add longer-distance links, using 7-port NNCs (or 8-porters, with two LANs, which they'd probably need anyhow).The regular periodic structure and even spacing simplifies calculations of channel re-use and routing. In particular, there are easy-to-calculate routes around dead nodes, so it can be re-configured on-the-fly by reasonably simple programs. Finally, the cost of the individual network nodes is proportional to the size of the community they serve, and thus the cost is spread evenly over the community members.

How about it?Sound interesting?

RESPONSES

#1 Fred Steiner:
Question, Mike: Would each RF link be on the same frequency, operating half-duplex, or each on separate frequencies? What I'm getting at is how essential to network efficiency as you propose it is use of separate RF channels, and how important is Full vs Half duplex operation? When I say "same channel," I mean each link transmitter and receiver on the same frequency, so each is limited by the activity of the others in the Half-duplex mode. Would this effectively do in your system, especially if file transfers, or other data intensive activities were going on between two parts of the honeycomb or could the system tolerate such if a single frequency is used for the links. If a single frequency is too inefficient, how many different ones would be needed for optimum efficiency?
wa9hdd

#2 Michael McClary:
Let's see if I've got this correctly. You're asking:

1) Would using the same frequency for transmit and receive on one long-haul link degrade the performance of this geometry?

2) Would using the same frequency for all outbound, and perhaps a common frequency for all inbound, long-haul links degrade the performance?

3) What frequency allocations would be required for the optimum operation of this scheme?

Numbers one and two are not really matters of geometry. They are matters of the technology of the individual links. You can't share one frequency half-duplex for a long-haul link if you're going to operate all three links to a node in the same band. This is because the timing of the incoming messages on the other links is not under your control, and any that come in while you're transmitting on that end of the band will be lost to receiver quieting caused by your outgoing signal. Thus the incoming and outgoing signals must be a channel-pair, on opposite ends of the band, and this means that you may as well run them both continuously.

With regard to sharing the frequency between long-haul links to different sites, there are several ways this could be accomplished, with varying impact on throughput.

The simplest approach would be to allocate three channel-pairs for the long-haul links (N-S, SE-NW, and SW-NE), and re-use them over and over. You might also want to allocate another three pair for the very-long-haul bypass. (A third set might be useful for cases where the next hunk of the net isn't far enough over-the-horizon for isolation, or where there is some local interference, but this shouldn't really be necessary, because these are fixed stations, and you can adjust the transmitter output to make the capture of the long-haul link receivers solid.)
With this simple approach you only need three pair for the basic grid. If you look at the previous item, you'll find that at least four pair are already allocated for the square-layout net.

You could use one outbound frequency, and time-multiplex it between the three neighboring sites. This would reduce your output capability by a factor of three. Performance would probably degrade slightly less than that, because your load would not normally be balanced between the three directions. There would still be the problem, though, that you would have three times as much potential for incoming data as for outgoing, and this would degrade the usage of the incoming frequencies by this same factor of three (which shouldn't hurt, because they'd be talking to three nodes, too). (In a network like this, traffic THROUGH your node will normally be so much greater than local traffic TO/FROM your node that the latter can be neglected.)

Remember, though, that unlike the LAN, the long-haul links are between single fixed stations which are line-of-sight (just barely) and separated by (roughly) 120 degrees. This means you can use fixed, directional antennas. An antenna with two very sharp nulls at 120 and 240 degrees from the major lobe is almost trivial to construct: two driven elements, correctly spaced and phased. Add a third (forming a triangle) and the correct phasing harness, and you have three such antennas in one unit. (Also, with three elements, you can move the angles of the two nodes independently, to compensate for the real-world differences in angle between the cities.) Using such an antenna (and perhaps transmitters sharing a master oscilator) you can use the same frequency pair simultaneously in all three directions, build the network with only one set of long-haul frequencies, and experience no degradation.

The LAN you use for accessing your local node should be at the same end of the band on which it is transmitting (so it doesn't quiet its long-distance receivers) or perhaps in the middle of the band. You don't want to have adjacent cities using the same frequency, because this would cause trouble in the fringes, where stations would hear and be heard by two or three network nodes. This requires three frequencies, placed like this:

```
1     2     3  12     3     1

   3 1     2     3   1 2
```

```
1      2      3  12      3      1

   3 1      2      3   1 2
```

Another set of three frequencies, at the other end of the band, could be used if the intermediate nodes also were to establish LANs, producing overlapping coverage.

```
1      2      3  12      3      1
   c a      b      c   a b
   3 1      2      3   1 2
a      b      c  ab      c      a
1      2      3  12      3      1
   c a      b      c   a b
   3 1      2      3   1 2
```

Placing the single-frequency LAN at the same end of the band as the network node's transmitters means that the node's LAN receiver is quieted, and this reduces its range. It might be better to place it at the other end, and raise the strength of the adjacent node's transmitters, to compensate for quieting in the long-haul net. (I think the original approach is better, though, because it will tend to push local activity away from the band end where it would interfere with the long-distance links. It also tends to restrict the LAN to the network node's vicinity, (because the network node will shout down stations it doesn't hear, and they'll eventually give up and move to another frequency) and prevents interference by stations working the LANs at more distant nodes.)

The local access LAN could, of course, be on a different band than the long-haul links, and this could eliminate the quieting problem.

Does this answer your question?

#3 Michael McClary:
It might be nice to experiment with another sort of local-access LAN with this setup (though this experiment wouldn't preclude operating a LAN with the protocols handled by the current TNCs simultaneously). This would be a two-frequency, full-duplex LAN, with the network node serving as a hub, transmitting and receiving at the same ends of the band as it does for the long-haul links, (with the local stations using them in the opposite sense), repeating messages and dealing out timeslots in its inbound channel. This would allow much higher utilization of the frequencies, since there would be no time wasted to collisions and collision-avoidance waits, and would prevent the throughput-drop that collision-allowing protocols experience when trafic increases, causing trafic-jams.

The protocol could be a token-passing style (with a slot reserved for requesting a position in the ring, which is less disruptive to existing virtual calls than the "reconfiguration burst" approach), or any of several other approaches. (Stations could still digipeat in, by means of transactions where a local station, on receiving the token, passes it to its out-of-range-of-the-hub partner (thus causing the hub to shut up), then relays what the partner sent.)

#4 Fred Steiner:
Good, now let's isolate some aspects of the newtwork for discussion, rather than trying to deal with it all at once. In particular, I would like to focus upon the interactive nature of the links in this system.

I will first try to define more clearly what I mean by half-duplex and full-duplex links:

Half-duplex -- Transmitter and receiver on each end of the link on are the same frequency. As with present TNCs and proposed NNCs, data carrier detect is used to prevent collisions. When either transmitter is active, its co-

is used to prevent collisions -- when either transmitter is active, co-located receiver is muted, conversely, when a receiver hears a data carrier, the TNC will not key the transmitter.

Full-duplex -- Transmitter and receiver are on different frequencies sufficiently far enough away to prevent desense (not quieting) of receiver by transmitter. The transmit freq of one side of the link is the receive freq of the other. Transmitter and receiver on both ends can operate simultaneously without interaction or degradation.

Let me use "inter-LAN" for your term "long-haul", and reserve "long-haul" for what you call "long-haul bypass", or in effect, special trunk links for high-volume traffic, perhaps at higher baud rates.

Also, unlike present protocol, acknowledgement frames would be sent hop to hop, instead of end to end, so whether a link were trying to communicate with a neighboring node or merely digipeat through it to somewhere else, the interaction would be similar, with a certain number of transmitted frames acked by the receiving node.

In this network scheme then, if LANs were on enough different frequencies to prevent overlap and mutual throughput degradation and on a different band than where inter-LAN linking would take place, the half-duplex inter-LAN links could share the same frequencies in a "time-domain multiplex" scheme (to use jargon with which you are much more familiar than I, so pardon it!). This is in fact similar to the present use of one frequency [145.01 MHz] for almost all LANs. Omni-directional antennas could be used at each node to hear well in all directions (and keep equipment costs low -- one transmitter, one receiver, one antenna).

To do so means that any distant inter-LAN transmitter [A1, for example] that can be heard by a given node receiver [B1] will degrade its throughput, both because it may hear collisions with other inter-LAN transmitters [C1 or D1] whose on-site receivers cannot hear A1 sufficiently well enough to detect data carrier [DCD from now on] and inhibit transmitting at that moment. Then would then lead to retries by all three transmitters until one could successfully be heard without collision by B1 and be acked. This series of retries would itself further degrade the throughput of not only the target node B1, but also all neighboring nodes able to hear A1, C1 or D1.

Use of significantly higher baud rates for the inter-LAN links over that of the LAN would be essential for the system to function at all, since shorter frame transmit times would tend to reduce the chance of collisions, as long as the ratio of TXD flag to frame data could be kept similar to that achieved at slower baud rates. However, as the system grew into a dense honeycomb, a threshold of overload would be very easily achieved, especially with high-volume inter-LAN traffic (such as file-transfers between certain nodes) that was inappropriate for trunk-link bypass.

Does this seem like a reasonable summary of the implications for this network scheme of half-duplex linking with omnidirectional antennas on the same channel? Efficiency of system throughput is assumed to be very important, whether it be between two relatively closely located nodes or very distant ones.It is also assumed that fairly dense file transfers would be necessary at times when the system also needed to support low-density, "chatting" traffic, because it would not be possible to locate and alert chatting stations that system resources were needed for higher priority dense traffic.

Naturally, there are ways to improve the situation, some of which you have alluded to already, such as use of several directional antennas, or a mixture of separate transmit and receive frequencking approach.

We can come back to the obvious alternative, which I think is more of what you had in mind from your attempts to use several channel pairs and directional antennas, namely, what is more a frequency-domain multiplex approach. Before doing so, however, is there any way to modify the geometry envisioning links where the

of the network node arrangement to get around the limitations of time-domain multiplexing outlined above? After your response to this, I'll take up my questions about frequency-domain multiplex linking and its implications for how this network would perform..

#5 Michael McClary:
Ok. Terminology first.

Quieting: disappearance of hiss and interfering stations with stronger received signal on FM.
Desense: reduced input sensitivity in a receiver caused by a strong signal at a nearby frequency.

Right, I used "quieting" for "desensing". (Shows you that I never used FM back when I was still a ham.)

Let's call the links:

Short-haul: Communication between nearest neighbor NNCs.
Long-haul: Communication between distant NNCs.
Inter-NNC: Either of the above.

Inter-LAN is a tough one to apply to this grid, because some of the NNCs may not have a LAN, and the long-haul links are probably inter-LAN.

(By the way, it's time division multiplexing, not time domain. The latter would make sense, too, but the professors didn't choose it.)

Now to the network:

Yes, your description of what happens if you try to go with omnidirectional antennas for the short-haul links, and to a single transmit and single receive frequency is good, though it's a bit understated. Two additional problems crop up. First, you must have ALL the short-haul links on the same frequency (or all the majors on one and all the minors on the other, if you work with offset receive frequencies). This in turn means that not only do you have to have all your neighbors but one silent at any given moment, but THEY must have all but one of THEIR neighbors silent at any given moment, and so on. The problem of synchronization of transmissions propagates across the entire net, until the timing of a transmission on an east coast transmitter could affect the timing of one on the west coast. Second, a straightforward collision-retry scheme could tend to encourage more than one node to transmit at a given time, generating, rather than reducing, collisions. These effects lead to a situation much worse than the one you described.

Whatever else you do, you don't want to try to run the inter-NNC links on a collision-retry basis. Collision-retry protocols all have the TERRIBLE problem of increasing overhead with increasing traffic. The more data you have to send, the more collisions you have, and the slower the data is sent. Hams don't normally operate live QSOs on a collision-retry basis. Why should their electronic proxies? (Especially since collision-retry is such a horrible way to allocate a scarce resource.)

As to modifying the geometry, that would be hard. Bear in mind that I'm not proposing that the LANs and NNCs be placed in these positions to make the net organization easier. I'm pointing out that this is how the cities are already laid out, and putting the NNCs anywhere else means setting them up where hams are scattered or non-existent, rather than where they are concentrated. My proposal is to recognize that this "natural" layout exists, and to plan the interconnection of the "natural" node locations to take advantage of it.

(While we're at it, I'll quickly mention that I had not envisioned using a collision-avoidance protocol for the inter-NNC links. Instead, I was envisioning links where the signal (at the receiver input) from the

establishing links where the signal (at the receiver input) from the
"correct" NNC would be stronger than, and thus suppress reception of,
signals from other NNCs that happened to be on the same frequency. Thus
the protocol would only have to avoid other users of the frequency that
WEREN'T part of the net. More on this once the discussion gets to
frequencies and/or protocols.)

C2974 CC154 Skip Hansen (WB6YMH,2964) 1/ 7/86 1:54 AM L:22
KEYS:/JUNK FLOPPIES/

Lyle,

I have had a lot of experience with the Mitsubishi disk drives and
it's all BAD! We have used about 150 of the drives (about 2 years ago)
and they had severe problems with diskette seating. About one time
in ten when the diskette is inserted the centering cone does not seat
correctly, if you notice the problem you can just reinsert it, however
if the problem goes undetected the usual result is a bombed disk.

We have used well over 500 Shugart SA-465's with only a couple of problems.
The major difference I can see is that the cone comes down directly parallel
to the diskette on the Shugarts (read that Panasonic) where as the
Mitsubishi's have the cone comming down at an angle. By the way we
also evaluated a pair of drives and didn't see the seating problem until
the first production order came in. The diskette ejector mechanism also
seems to fire on diskette insertion unless it is adjusted properly which
makes the seating problem about 100 times worse.

I've seen the SA-465's for under $100 and would highly recomend them
even if they were twice the cost of the Mitsubishi's.

73's Skip WB6YMH

C2974 CC155 Lyle Johnson (WA7GXD,2973) 1/ 8/86 1:35 AM L:14
KEYS:/FLOPPY DRIVES/
A: 154

Gary,

Skip,

As I was saying before I got bumped off line (!), the newer Mistubishi
drives have a different door mechanism than the older "push-button" style
drives. I also have Fujitsu drives under evaluation (sampled loaners)
and Tandon TM65-2 half-height drives enroute. Do you have any experience
with these drives? The Shugart/Panasonics are uinfortunately only
available locally through Hamilton Avnet as authorized distributors, and
franbkly I have had nothing but trouble from Ham Avnet and would rather
not spend another penny with them.

Lyle

C2974 CC156 J. Gordon Beattie Jr. (N2DSY,2990) 1/ 8/86 12:17 PM L:35
KEYS:/SLIDING WINDOWS/
A: 145

Phil,
I re-read your comment on ACK-ACK protocols and I really must
dispute two important points.

The first point was that half-duplex protocols do not benefit from
sliding window protocols. YOU CAN`T BE SERIOUS !
If I send several packets at once and then take one ack then I am using the
channel more efficiently than if I send the ack after each packet.
You can test this by setting up two packet boards with MAXFRAME=1 and then
comparing it to the same setup with the boards set up with MAXFRAME=7.
This situation would also be further impacted if the rf path was through a

Phase III or IV sattelite. The users may or may not want to use the value of seven, but most use a value ranging from 2-4. Switches, however will probably use the maximum value of 7.
While I'm on the subject, stations should also use as long a PACLEN as possible This will also help reduce the number of "line-turn-arounds".

The second thought that you inspired was that we have a sliding-window ack ack style protocol in the combination of AX.25 Link Level and the X.25 Packet Level Protocols. Howie has implemented them and they are undergoing testing in Florida. Here's how it goes:
A sends an I-Frame containing a Level 3 DATA Packet to B.
B sends an I-Frame containing a Level 3 RR to A.
A sends B an RR-Frame to A completing the handshake.

This exchange could be extended by the transmission of other DATA Packets going from B to A.

On this one Phil you are way off base, ask anyone who has worked with Bisync over a satellite channel ! The delays of line turnaround gave rise to the popularity of sliding window protocols for terrestrial use too.
Check all the literature Phil; you'll find that most vendors are using some HDLC-style protocol in their proprietary and OSI software products.

C2974 CC157 Phil R. Karn (ka9q,2979) 1/ 8/86 3:36 PM L:52
(ORIG.) 1/ 8/86 3:14 PM L:45
KEYS:/LINK PROTOCOLS/

Sorry, Gordon, you misunderstood me.

What I was trying to explain is that in a half-duplex (i.e., radio) environment, sending X bytes divided up into N frames instead of just sending X bytes in 1 frame (i.e., decreasing the window to 1 frame and increasing packet size by N) isn't necessarily a good idea.

The only reason I can see for a sliding window protocol on radio is the fact that a hit on a single large packet will render the whole thing unusable, while a hit in a stream of smaller packets will render only that packet and all that follow unusable (because of the go-back-N retransmission technique). Now it depends on what your primary cause of errors is whether this makes any difference.

If errors are gaussian (i.e., each bit has an independent probability of being wrong) then the probability of successful packet transfer goes down as p^N, where p is the bit error rate and N is the packet size in bits. On the other hand, if most lost packets are due to collisions in a CSMA environment, they will occur only at the beginning of the transmission, during the "collision window" caused by propagation delay, keyup "blind periods" and so forth. In this case collisions will require you to retransmit everything anyway, so you might as well send everything in a single large frame instead of a bunch of tiny frames in order to reduce link layer header overhead.

The other major characteristic of a sliding window protocol, namely "pipelining", is not a good idea on a half duplex channel. You should not allow more than one transmission to be outstanding at any one time, because of the possibility of a subsequent transmission clobbering the returning ACK for the previous transmission. If you accept this point, then the only reason for the sliding window protocol could be the point I mentioned earlier about gaussian noise. However, there are clearly some better techniques than go-back-N that ought to be studied in this case (e.g., FEC).

Howie's protocol is probably NOT an "ack-ack" protocol. Yes, there is a three-way handshake, but the timers on each end (the one

is a three way handshake, but the timers on each end (the one
awaiting a link level ack for the original frame and the one on the other end awaiting a link ack for the level 3 RR) are equal. This does not buy you anything. In order for ACK-ACK to work, the ACK retransmission timer has to be shorter (by some integral number) than the data retransmission timer. I thought I made this clear in my earlier analysis.

The fact that LAPB is popular in commercial systems doesn't mean that it's automatically the best thing for the amateur radio environment, with its far higher error and loss rates. I think the answer lies in some rigorous analysis and simulation testing with reasonable assumptions. Simple appeals to authority, although one of your favorite tactics, doesn't work here.

Phil

C2974 CC158 J. Gordon Beattie Jr. (N2DSY,299Ø)   1/ 9/86   1:57 AM L:18
KEYS:/YOU CAN'T SHATTER THE WINDOW !/
A: 157

Phil,
I agree that you do need to go back-N if the first packet is clobbered, but you are forgetting that because we have increased the amount of data by expanding the window, we have eliminated one party from the next few rounds of contention. I do believe that this is one factor often ignored.

There is a second factor that we must address: Contention can be reduced on Network Trunks and Access channels by optimization of the following factors:

1. Sharp antennas
2. Minimum power
3. High frequency
4. Synchronized transmission
5. Minimum number of stations in an RF domain

Good network engineering will allow us to use more efficient protocols for data transmission.

Remember: We don't allow oxcarts on the Interstate !

C2987 CC247 Howard Goldstein (N2WX,2987)   1/ 5/86   5:Ø8 PM L:5
KEYS:/MIGHTY DEAR RAM/
A: 246

Lyle,

I could try to gen up a release if you don't mind being the first to run it! gator 2 is running with 32k ram equiv so it might work...

73 Howie

C2987 CC248 Tom Clark (W3IWI,2976)   1/ 8/86 11:5Ø PM L:8
KEYS:/1.1.2 BUG/DIGITAL LOOPBACK TEST DOESN'T WORK/INVOLVES JMP 1Ø/

Howie, this is to report a minor bug in 1.1.2. I put 1.1.2 ROM in a new Rev.2 board and found that it failed the Digital loopback test (involving JMP1Ø). With a 1.1.1 ROM everything worked fine. After tuning the TNC2 up, I re-installed the 1.1.2 ROM and everything worked fine (except for the JMP 1Ø test), so I'm sure it isn't the hardware. This TNC2 had R98 = 1ØØk per the latest "gotcha" from TAPR HQ. Haven't tried it on any others.

73, Tom

Gator 2 PAD Survival Sheet 1/5/86

1.Ø

The Gator 2 PAD (Packet Assembler/Disassembler) provides an alternate access method for users who do not support AX.25 level 3 in their TNCs - the PAD facility.

Up to fifteen pad users are supported simultaneously on ports A through O.

2.Ø Placing calls

2.1 Activating the pad

The pad is activated by connecting to the switch and sending a packet.

```
cmd:CONNECT N2WX-7
cmd:*** CONNECTED to N2WX-7
{4

gator 2 pad Ø31ØØ3Ø5724 port B

enter: call [, digi1 [, digi2 [, digi3] ] ]
to?
```

-- example of activating the pad --

2.2 Specifying the destination

Now the pad needs to know who you wish to call. Enter the callsign (along with any ssid) and press.

```
to?N2WX-1
```

In some cases the station you want to connect with will not be within direct range of the switch. Gator 2 supports both networked calls and calls using digipeaters.

```
to?K4OZS,ORL,OCF
```

Above is an example of what to tell the Melbourne switch to call K4OZS who lives beyond the range of the network layer (but is available this way).

```
to?K4NTA @3Ø5STU
```

Although this example is not supported yet, it will be should STU implement gator 2.

2.3 Indications

Once you've issued a callsign (and optional @/digipeater parms) one of three things could happen.

> Normally you'll get a message like
*** pad: connection reset
This means the call was succesfully completed. It is analogous to a "*** connected" message

You could get a *** pad: connection reset message if the person on the other end reconnects or somehow resets his own link.

> You get the prompt again

```
to?N2WX-1
```

enter: call [, digi1 [, digi2 [, digi3]]]
to?

If this happens the person you called is already using the pad on a call with someone else. Try again later.

> You get disconnected ("kicked off")

The person at the other end had CONOK off, was busy, disconnected on his own accord, or the attempt to call failed by retrying out. You'll get kicked off anytime a pad call goes down.

M 31 Phil R. Karn (ka9q,2979) 1/ 9/86 3:13 AM L:6
KEYS:/TWO PORT KLUDGES/
TO: KA6M, ka9q

Hank, how about a REAL "two port digipeater"? How about 4 ports? My IP/TCP code is in late stages of development, in that I have an IP switch module running under the TCP code (which is optional) it can switch packets NOW, all it really needs is a routing algorithm.

Phil

C2974 CC159 Phil R. Karn (ka9q,2979) 1/ 9/86 3:11 AM L:4Ø
KEYS:/WINDOWS AND PERFORMANCE/

Gordon,

My assumption was that the amount of data in a transmission remains the same; the question is whether it should be sent as a bunch of tiny back-to-back frames (with attendant overhead but a chance to salvage at least the beginning of a transmission) or as one big frame.

I am hacking away on a simulation to get some numbers on this question. What I am doing is computing the expected number of usable data bits in a transmission (i.e., the number of data bits received up until the first hit, which renders the remaining frames unusable) as a function of BER and the number of frames in the transmission. I've just put that function inside a brute-force search loop that for each possible number of frames in the transmission (1-7), find the optimal total transmission length. This takes into account several conflicting parameters:

1. The fewer the frames in the transmission, the less header overhead there is.
2. The more frames in the transmission, the greater the chances of salvaging something at the beginning of the transmission should there be a hit towards the end.
3. The larger the total transmission, the less overhead is wasted on acks.
4. The smaller the total transmission, the less is wasted on unusable data (data sent after the first hit).

What I'm finding is interesting. I'm using overhead = 16Ø bits/frame, ackcost = 16Ø bits/ack, and BER = 1Ø^-4. For each window size, there is an optimum transmission length and it increases as the window size goes up. HOWEVER, the peak efficiency goes DOWN with increasing window size. Here are the raw numbers, I haven't double checked them yet:
1 frame: optimum length 1711 bits, peak efficiency = 69.3%
2 frames: 28Ø3 bits, 63.9%
3 frames: 3647 bits, 59.7%
4 frames: 4359 bits, 56.2%

5 frames: 4984 bits, 53.2%
6 frames: 5543 bits, 50.6%

Phil

C2974 CC160 Phil R. Karn (ka9q,2979) 1/ 9/86 2:29 PM L:33
KEYS:/MORE ANALYSIS RESULTS/

I've been playing some more with my analysis model and my suspicions are confirmed.

For a given transmission length, there is an optimum number of frames to divide that transmission up into. As the transmission length increases, that optimum number also increases. However, the efficiency (where efficiency is defined as the number of USABLE data bits that arrive at the receiver divided by the total number of data, header and ack bits sent on the channel) decreases. It turns out that for larger messages, you win by dividing them up into a number of smaller single-frame transmissions. In no case did I find that a window size greater than one improved efficiency.

Assumptions: gaussian bit errors (this is the case I wasn't certain about; a CSMA collision envirqonment would seem to benefit more from single-frame transmissions), the overhead on each frame is 160 bits (one AX.25 frame), the overhead of an acknowledgement is also 160 bits, and the bit error rate is on the order of 10^-4. I also tried 10^-3 and 10^-5, and while it affected the actual numbers, the relative merit of single-vs-multi-frame transmissions didn't change. I also assumed an ACK-ACK protocol, where there were no unnecessary data packet retransmissions due to lost acks (but I did weight the cost of sending acknowledgments by the probability that the fixed-length ack makes it back). This should not affect the relative results in the LAPB (non ACK-ACK) case because everything should simply decrease by a constant factor based on the fixed ack transfer probability at a given BER.

If anyone would like to experiment with this program (or verify my math) I'd be glad to upload it.

Phil

C2974 CC161 Lyle Johnson (WA7GXD,2973) 1/10/86 1:00 AM L:37
KEYS:/NNC PROGRESS REPORT FOR 09 JAN 1986/

To: Interested NNC Followers
Fm: TAPR
Re: NNC Prototype Testing Update

The NNC Prototype PC board has now been suitably hacked due to
a) incompatibility of HD64180 uP with Z80(tm) peripherals running Mode 2 interrupts
b) requirement of parallel printers to get a pulsed signal telling when data is ready.

Tonight, I patched the Z-system BIOS to recognize the modified Z80 PIO B port as the system listing device. It is now fully Mode 2 interrupt driven and the OkiData uLine 93 printer loves it! The wait-state generator has been reworked to provide a single wait-state during op-code fetches in the lower 256k of address space as well as providing the required leading edge delay in the !LIR signal from the HD64180 to more closely emulate the !M1 signal from a standard Z80.

The system clock is 4.608 MHz (9.216 MHz crystal) and the BIOS has been modified to acccept this data rate. The disk drives have been interfaced and run just fine with the disk controller running under DMA control.

The SIOs have been tested using a simple test procedure sending HDLC frames and receiving them in a full-duplex loopback system; interrupts have not yet been tested with the SIOs. The SCSI interface has been tested to the extent of verifying that the uP can in fact talk to the registers of the NCR5380 chip and read the registers from the chip.

The modified schematics will be sent to St Louis this weekend to get the necessary changes CADded. We hope to have the revised artwork back in Tucson by Friday the 17th and test boards in time for the annual meeting. One additional change being implemented is replacing the op-amp RS232 driver with a CMOS rs232 driver for better operation at higher data rates.

If the revised artwork is accurate, then we will immediately push the button on getting the Alpha and Beta PC boards fabricated.

Onward!

C2987 CC249 Howard Goldstein (N2WX,2987) 1/ 9/86 10:11 AM L:5
(ORIG.) 1/ 9/86 10:06 AM L:3
KEYS:/1.1.2 DIGITAL LOOPBACK WORKS FINE AT N2WX/
A: 248

Not sure why yours is not working. Make cure carrier detect is not true on the SIO (1.1.2 DOES have a prob vis-a-vis full duplex but only as it relates to carrier detection...it's now too friendly to other channel users). 73 Howie

M 461 Howard Goldstein (N2WX,2987) 1/10/86 8:53 AM L:2
KEYS:/PROGRESS ON DUAL PORT TNC2?/
A: 30 TO: KA6M, (Group 95)

Sorry Hank my stuff isn't ready for two sync ports yet. 73 Howie

M 908 Rich Amundson (WA0JFS,2965) 1/11/86 11:06 PM L:14
KEYS:/SOCKETS FOR 64180/
TO: (Group 95)

TO: GROUP 95 AND LYLE JOHNSON
FROM: WA0JFS RICH AMUNDSON
RE: 64180 SOCKETS

Dave Huffman KA0DNB has acquired a pair of 64180 chips for experimentation and is in need of sockets for same. If anyone here has a source for them please leave me a note here so I can pass a long the info. He says they are apparently only made by Hitachi as the chips and are of the type DP64S.
Thank you.

Rich Amundson WA0JFS

C2974 CC162 Lyle Johnson (WA7GXD,2973) 1/12/86 7:14 PM L:40
KEYS:/NNC MODEM TESTING PROGRESS/

To: All
Fm: TAPR
Re: NNC Testing Update

Today, 12 January 1986, the prototype NNC Modem Board has undergone initial testing. This board contains four (4) XR2206/XR2211 modems, a crystal-controlled baud rate generator, four (4) state machines for NRZ <--> NRZI conversion and a tuning indicator.

As might be expected in an initial prototype, there are a few errors in the PC board layout. After correcting these errors, the following emerged:

1) All XR2211 demodulators work. The residual noise at the inputs

to these chips is on the order of 2 or 3 millivolts. With an input signal of under 2Ø mV they lock solidly.

2) The baud rate generator works fine. It is based on a 4.9152 MHz crystal oscillator. All output leads are bypassed with a series 1ØØ ohm resistor and small capacitor to ground (L network).

3) All XR22Ø6 modulators work.

4) All state machines work, clock recovery is solid with a signal of 2Ø mV to the associated XR2211.

5) The tuning indicator portion was not tested.

In light of the reported difficulties with TNC 2 oscillator RFI, the amodem board was tested with a Kenwood TR26ØØA HT (includes S-meter). Unless the capcitive end of the antenna was physically positioned less than 1/4" from the PC board, it was impossible to hear any spurious signals from the board. This is with squelch open.

Thus, the initial state of the Modem board for the NNC is favorable, with rfi very low and modem sensitivity very high. Further testing will be conducted in the near future and reported here.

The above testing was conducted by Eric Gustafson, N7CL, designer of the modem board.

M 1628 ANDY FREEBORN (NØCCZ,2983) 1/14/86 7:13 AM L:14
KEYS:/LOOPBACK TESTING/
TO: (Group 95)

Paul,
Reference Gwyn' CC25Ø and the problem that Fred, AI6C, was having with repair of the returned TNC2REV2.
.
He, in spite of your suggestions could not get pin 19 to go high in order to continue the digital loopback test. All chips on the sick board had been substituted from an unbuilt TNC2REV2 of mine with the same results continuing.
.
He then installed the headers, pin 19 behaved properly and the machine worked perfectly. It has been on the air since Sunday and going fine. It will be returned to its owner today.
.
Andy

M 1789 Howard Goldstein (N2WX,2987) 1/14/86 2:23 PM L:7
KEYS:/DIGITAL LOOPBACK/FULLDUP/
TO: WA7GXD, ad7i, WB9FLW, NK6K, W1Bel, NØCCZ, (Group 95)

Despite the setting of FULLDUP, 1.1.2 will continue to honor DCD. THIS IS A SOFTWARE ERROR and will be corrected in the next release. Thanks Harold for pointing this out.

Perhaps the digi loopback has been failing because of this (as Andy's msg to Paul implied.)

73 Howie

M 1932 Pete Eaton (WB9FLW,297Ø) 1/14/86 8:45 PM L:17
KEYS:/KANTRONICS ANNOUNCES VER 2.Ø FIRMWEAR FOR KPC/
TO: (Group 95)

To: All
From: Pete WB9FLW
Subject: Kantronics Ver 2.Ø Firmwear

Dist: Open

Kantronics has announced the availability of Version 2.Ø software for the Kantronics Packet Communicator. New features are:

* AX25L2V2 Protocol

* Multiple Connects

Price for the upgrade is $2Ø.ØØ and will be available February 3rd.

C2987 CC25Ø Gwyn Reedy (W1Bel,2975) 1/14/86 5:58 AM L:15
KEYS:/LOOPBACK TESTING/

Howie, I have received several reports from TNC-2ØØ builders about the failure of the digital loopback test. All reports seem consistent. Have observed the problem on one board that was carried to the house for diagnosis also. Am telling everyone not to worry, that it works fb on the air...hope that continues to be true. TAPR has 5ØØ of those ROMS out there, but the discxovery rate on the problem will be slower since those folks are already up and running, and may not do a full checkout after modifying their boards. Am not anxious to do an immediate retrofit of ROMS unless there is an on- the-air impact. Can fix the bug with the next upgrade (What else can you possibly add to such a rich command set???). Lets talk about getting you a rev 2 board. (Surprised you don't have one already.) P.S. We are providing the 1ØØk for R98 so that is not the source of the loopback problem. 73, Gwyn.

C2987 CC251 Howard Goldstein (N2WX,2987) 1/14/86 3:Ø4 PM L:3
KEYS:/DIGITAL LOOPBACK/RELATED TO FULLDUP?/
A: 25Ø

Gwyn - if it is related to the FULLDUP 1.1.2 problem, it can be fixed by pulling false the DCDA* input. 73 Howie

C2987 CC252 Tom Clark (W3IWI,2976) 1/15/86 1:1Ø AM L:2
KEYS:/DIGITAL LOOPBACK/1.1.2 PROBLEM EXPLAINED/

Howie, the FULLDUP symptoms seem to account for my digital loopback problem I reported on earlier. Looks like another gotcha!

M 2155 Thomas A. Moulton (W2VY,995) 1/15/86 1Ø:45 AM L:2
KEYS:/KANTRONICS ANNOUNCES VER 2.Ø FIRMWEAR FOR KPC/
A: 1932 TO: (Group 95)

I wonder if the multiconnect interface is like anything we've seen before, or is it again their own features?

M 3Ø31 Pete Eaton (WB9FLW,297Ø) 1/17/86 1:57 PM L:6
KEYS:/FANTASTIC/
TO: (Group 95)

Gwyn:

Just got the latest PSR, FANTASTIC! If you keep going like this by next year it will be as thick as the 1986 ARRL Handbook! (ok half as thick as the 1986 Handbook).
Pete

C2987 CC253 Tom Clark (W3IWI,2976) 1/17/86 2:17 AM L:7
KEYS:/ANOTHER 1.1.2 BUG/OR IS IT A FEATURE?/

Howie, apparently we found another 1.1.2 gotcha: You can loop back and connect to yourself thru one digi, but not more. e.g.
C W3IWI V WB4JFI-5 works, but C W3IWI V WB4JFI-5,WB4APR-6 does not.
Perhaps this is an un-advertised feature to prevent people from cluttering the airwaves by connecting to themselves?

73, Tom

M 3383 Pete Eaton (WB9FLW,2970)   1/18/86 10:58 PM L:11
KEYS:/FCC APPROVES NEW DIGITAL PRIVILEGES!/
TO: (Group 95)

To: All
From: Pete WB9FLW
Subject: FCC Approves New Digital Privlileges

Part 97 of the FCC Rules & Regulations have been amended to allow unattended automatic control of Amateur Digital Communications above 50 MHZ. PR Docket 85.105 was approved by the Commission on Tuesday, January 14th. It will allow digipeating and other Amateur Digital Repeating Devices to operate unattended.